N. 2 - GIUGNO 1988
L. 14.000
AMIGA
BYTE
68.000
by Elettronica 2000
Sped. in abb. post. Gr. III/70
COMPUTERSOUND
PROGETTO
MIDI
GRAFICA
IN 3D
AVVENTURE
RECENSIONI E
SOLUZIONI
ASSEMBLER DATA
SOFTWARE
EXPRESS
TIPS & TRICKS
VOLARE DAVVERO CON
FLIGHT SIMULATOR II
COME VINCERE
LA GURU
MEDITATION
CORSO
DI BASIC
LE VARIABILI
NEL "C"
DOS
LE DIRECTORY
AMIGA
by KATHERINE MAHONEY

Oberheim
Oberheim. Matrix-1000
Select
Patch
Channel
Fine Tune
Units
Data Dump
Ext Funct
Bank Lock
999
Enter
Volume
Power
- Expander a 6 voci - Rack 19"/1 U.
- 1000 suoni analogici (800 ROM - 200 RAM)
- 36 voci polifoniche sfruttando il Group Mode
- Spillover abilitato
Matrix-1000
distribuzione esclusiva: MEAZZI s.p.a. 20161 milano- via bellerio 44 - tel -02-6465151-telex:335476

N. 2 - GIUGNO 1988

Direttore
SIRA ROCCHI

Direzione Editoriale
MARIO MAGRONE

Direzione tecnica
ENRICO DI ZENOBIO

Segreteria di Redazione
SILVIA MAIER

Grafica
NADIA MARINI

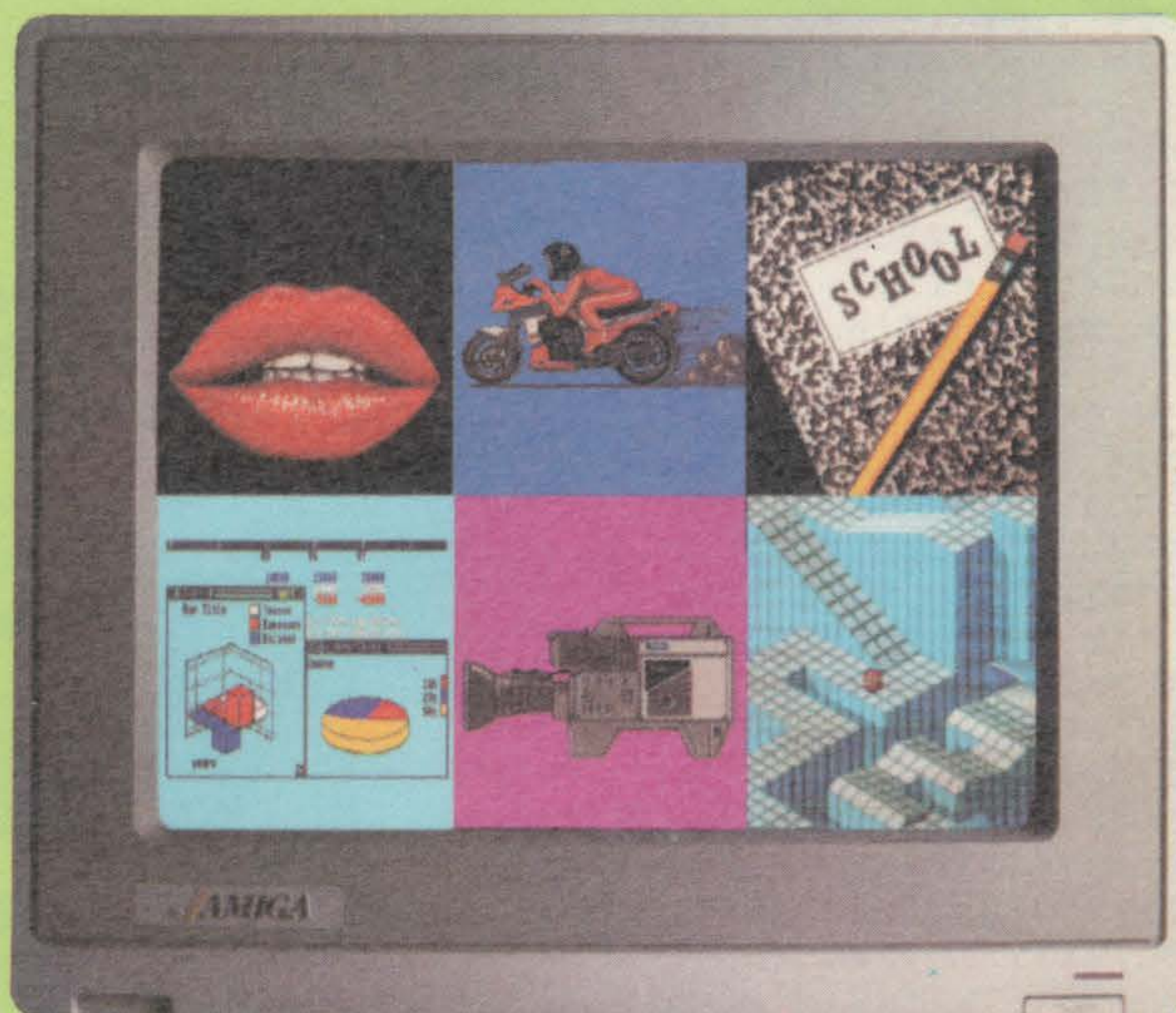
Fotografie
MARIUS LOOK

Disco a cura di
VITTORIO FERRAGUTI

Hanno collaborato: Giulio Bonifazi, Alessandro Bossi, Luca Brigatti, Marco Brovelli, Carlo Cattoni, Enrico Donna, Leonardo Fei, Maurizio Mobilio, Rolando Poliziani, Ricky Sword.

Redazione
C.so Vitt. Emanuele 15
20122 Milano
tel. 02/706329

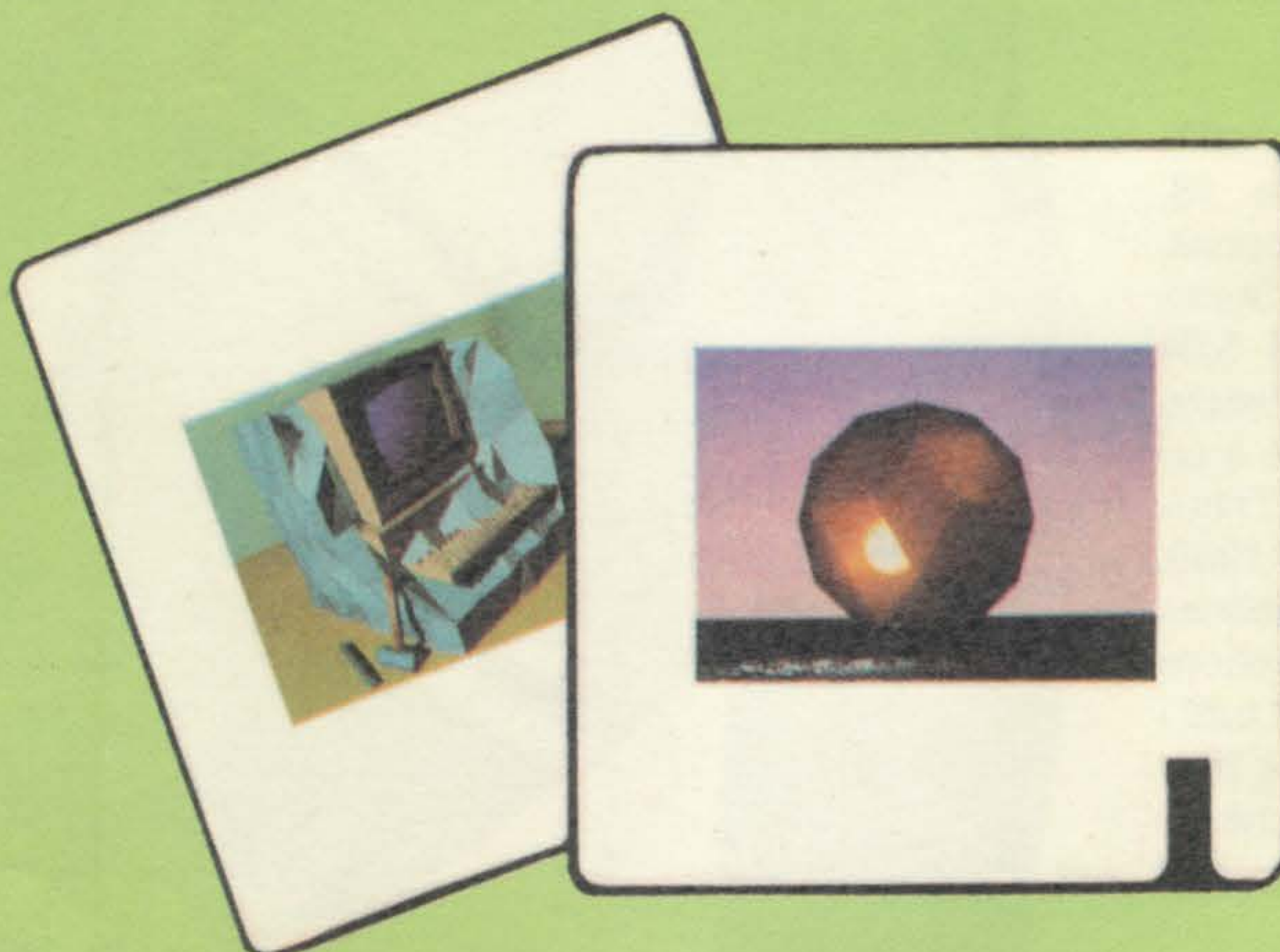
Amministrazione, Redazione, Pubblicità, Arcadia srl: C.so Vittorio Emanuele 15, 20122 Milano. Fotocomposizione: Composit, selezioni colore e fotolito: Eurofotolit. Stampa: Garzanti Editore S.p.A. Cernusco S/N (MI). Distribuzione: SO.DI.P. Angelo Patuzzi spa, Via Zuretti 25, Milano. Amiga Byte è un periodico mensile registrato presso il Tribunale di Milano al n. 215 il 29 marzo 1988. Resp. Sira Rocchi. Spedizione in abbonamento postale Gr. III/70. Pubblicità inferiore al 70%.

IL MENU

Modellazione 3 D

Un viaggio affascinante, quello nella grafica, che prenderà le prime mosse da quella bi/tridimensionale modellando la figura di un cubo con i migliori pacchetti grafici esistenti.

di MAURIZIO MOBILIO

Inizia qui il nostro viaggio attraverso la grafica bi/tridimensionale attraverso il quale cercheremo di sfruttare tutte le possibilità che Amiga ed i diversi pacchetti grafici ci mettono a disposizione. Questa macchina opera nel campo professionale sin dai tempi della sua prima apparizione (Tandem, Musica è, Baby Show, Jeans, Fuori Orario, etc.) e riuscire ad ottenere, a livello amatoriale, gli stessi risultati del campo grafico professionale è tutt'altro che impossibile, anzi! L'unico inconveniente che incontreremo, ma più avanti nel viaggio, potrà essere quello inerente all'animazione tridimensionale, visto che questa richiede hardware aggiuntivo. Nell'ambito della generazione della grafica computerizzata tridimensionale, esistono diverse fasi da attraversare prima di ottenere il risultato finale, ovvero un'immagine completa di tutte le sue componenti di forma, colore, profondità ed ombre; ovvero la modellazione, la struttura delle luci e del punto d'osservazione, il rendering ed, eventualmente, l'animazione.

Il primo passo, quello di cui ci occuperemo questa volta, riguarda la modellazione degli oggetti, fase basilare che definirà la struttura generale dell'immagine in base alle sue proprietà fisiche; in parole povere ci occuperemo di disegnare lungo i tre assi cartesiani gli oggetti che costituiranno la nostra immagine finale. Ciò potrà apparire inizialmente un po' complesso non tanto per una reale difficoltà operativa, quanto perché siamo abituati a pensare in maniera bidimensionale, cioè tutto ciò che solitamente si disegna (ad esempio con il Deluxe Paint o con il Photon Paint) viene posto su di un unico piano e l'eventuale prospettiva viene sviluppata sempre sullo stesso piano. Nell'ambito del tridimensionale dobbiamo abituarci a pensare che un oggetto ha anche una sua profondità. Per capirci meglio, facciamo l'esempio di un cubo (questo esempio, manipolato in vari modi, ci seguirà per tutto l'attuale percorso). Quando lo disegnamo in maniera tradizionale il lato che si sviluppa nel senso della profondità viene rappresentato come un quadrato deformato. Nel nostro caso,

invece, dovremo disegnare quel lato identico a tutti gli altri; ciò che costituirà differenza sarà la sua posizione. Vediamo però di essere un po' più pratici. Innanzi tutto diciamo quali sono i pacchetti disponibili sul mercato che contengono un modellatore che operi in modo da generare dati intercambiabili direttamente o indirettamente e quali sono, in maniera stringatissima, le sue caratteristiche, tenendo conto che importante è la possibilità di generare oggetti che possono essere utilizzati da più programmi, in modo da poter sfruttare le caratteristiche dei vari pacchetti poiché, sino ad oggi, non ne esiste nessuno, per Amiga, che permetta, lui da solo, una manipolazione delle strutture in maniera professionale e completa. I pacchetti, dunque, sono:
Rot 3D della Aegis,
EGG 1 della Aegis,
OCT 1 della Aegis,
Sculpt 3D della Byte by Byte,
Interchange della Syndesis,
Fot Pubblico Dominio.

IL PACCHETTO VIDEOSCAPE

Il Rot 3D della Aegis fa parte del pacchetto Videoscape 3D ed è un modellatore di uso alquanto semplificato ma anche, purtroppo, abbastanza limitato; è utile per generare oggetti di piccola dimensione composti da un numero limitato di poligoni.

La struttura fornisce le tre viste tipiche, frontale, superiore e laterale; un palette a soli 16 colori (Rot, non dimentichiamolo, nasce per operare con Videoscape e comunque in questa fase non ci interessa quali colori verranno dati all'oggetto, in quanto li genereremo successivamente senza curarci di quelli eventualmente presenti) ed i riferimenti alle posizioni spaziali.

La costruzione dell'oggetto viene effettuata sia tramite l'uso del mouse che direttamente con parametri spaziali inseriti da tastiera. Il numero di punti generabili è limitato a 99, così come quello dei poligoni (limite teorico).

È consigliato, data la velocità d'uso, per generare piccole strutture particolari che verranno poi rimaneggiate da modellatori più evoluti ed inserite in archivi. Attenzione a non confonderlo con la versione di Pubblico Dominio, che ha la stessa struttura grafica e lo stesso nome ma che genera solo file in un formato suo particolare, utilizzabile quindi esclusivamente nel suo spazio circoscritto.

Sempre nell'ambito del pacchetto Videoscape 3D troviamo l'EGG, ovvero un generatore di primitive geometriche abbastanza completo. Oltre che le primitive classiche, EGG permette di generare strutture frattali definite matematicamente solo per quanto concerne i limiti estremi, operando definizioni sui punti intermedi tramite una generazione pseudo random.

L'uso di questi limiti estremi si rivela utile per definire oggetti con superficie rugosa o sfondi montagnosi.

Altra struttura parzialmente random è quella relativa alla generazione di spazi stellati. Ogni primitiva permette diverse varianti strutturali rendendo questo generatore indubbiamente utile.

L'OCT non è un modellatore ma un manipolatore di modelli spaziali; la sua funzione è quella di modificare le proporzioni di un oggetto, la sua posizione relativa o il suo colore basilare. Oltre che questo, permette di unire più primitive elementari per formare oggetti complessi.

Il modellatore dello Sculpt 3D è indubbiamente il più sofisticato e quello che, probabilmente, viene utilizzato dai più nella fase finale della modellazione.

I PASSAGGI DI FORMATO

L'Intechange è un convertitore di formato molto elastico e versatile grazie al quale si possono effettuare tutti i passaggi di formato dei quali parleremo in seguito; attualmente opera con le strutture dello Sculpt 3D e del VideoScape 3D, ma presto sarà possibile operare anche con altri formati, e quindi con altri programmi, grazie alla semplice aggiunta di alcune librerie di sistema.

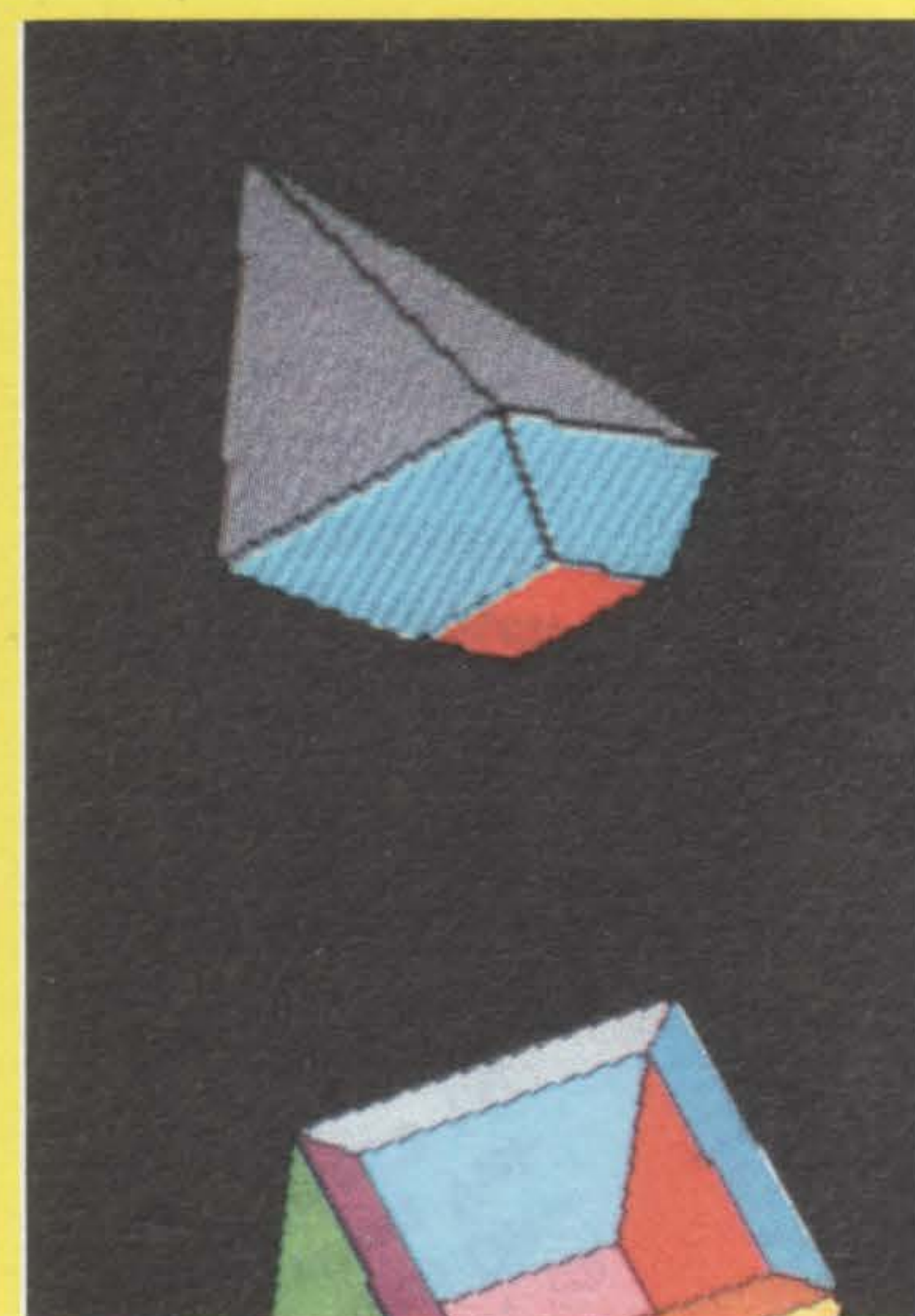
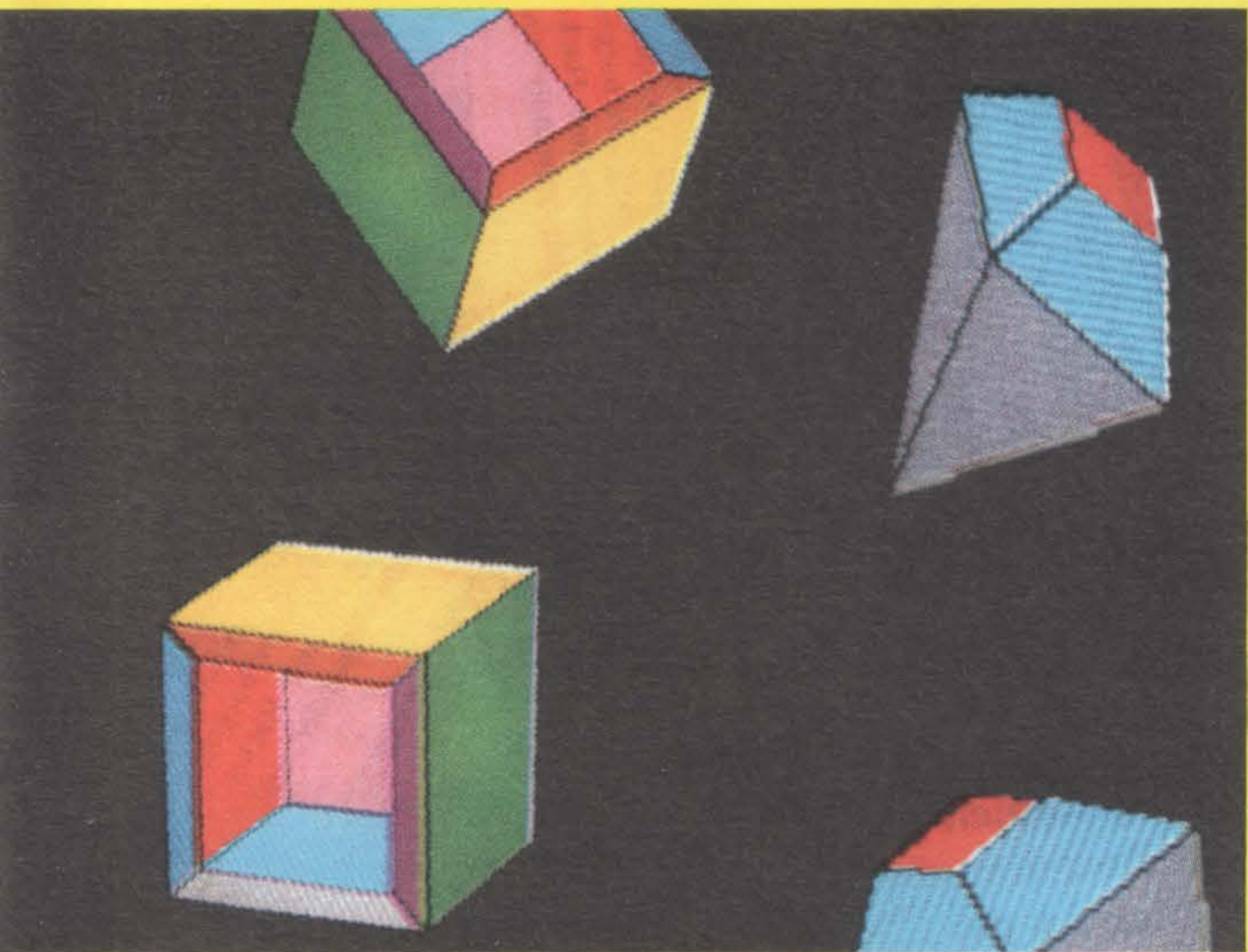
Il Fot è un convertitore di formato disponibile nel circuito del Pubblico Dominio (ha, quindi, il grande vantaggio che non deve essere acquistato) il cui limite è quello di operare in maniera univoca, ovvero dal formato Sculpt 3D a quello VideoScape 3D.

Vediamo ora come si può procedere in base alla struttura di più oggetti. Nel caso di primitive, la strada da seguire è ovviamente la più semplice, ovvero generarle con uno qualsiasi dei pacchetti, ed è inutile qualsiasi esemplificazione particolare visto che basterà scegliere da menu la primitiva che si desidera generare, dopo di che sarà il programma stesso ad occuparsi di tutto il lavoro. Il modo più semplice per ottenere un veloce apprendimento è quello di mantenere, sin dall'inizio, un esempio che ci accompagni lungo tutto il percorso ed è quello che faremo in questo ed in tutti gli altri articoli che seguiranno su argomenti inerenti. Nel caso si voglia generare, ad esempio, un poligono con un texture marmoreo (con il termine texture o texture mapping si intende la struttura estetica dell'oggetto, ovvero la sua immagine) e con una superficie rugosa, la prassi potrà essere la seguente: tenuto conto delle proprietà di una superficie rugosa, cioè la non linearità della superficie e quindi la presenza di ombre, ed in particolare ricordando che attualmente non esiste un modellatore in grado di generare in prima istanza un texture simile, bisogna pensare alla fase di rendering anche durante quella di modellazione. Innanzitutto vanno decise la dimensione e la forma generale del poligono (in questo caso prenderemo come assunto che si tratti di un cubo) e quella massima relativa ai rilievi. Tramite il generatore di primitive

Diversamente che nel disegno tradizionale, il lato del cubo che si sviluppa nel senso della profondità è identico a tutti gli altri.

EGG creeremo una superficie frattale il cui limite massimo dei rilievi sia, in percentuale, alquanto basso rispetto alla dimensione generale (un 5% rappresenta già una percentuale massima), in modo da ottenere dei rilievi che siano delle semplici rugosità. Va precisato, prima di proseguire, che per ottenere l'effetto che abbiamo deciso il programma sfrutterà quelle forme geometriche definite frattali, ovvero una serie di poligoni (triangoli) costruiti ognuno avente i lati in comune con quelli adiacenti, disposti in modo tale per cui la superficie dell'oggetto generato non sarà lineare, ma riporterà delle variazioni di livello.

CON IL PROGRAMMA EGG

Per ottenere questo basterà caricare il programma EGG e, quando apparirà un menu di scelte, selezionare la numero 9. Verrà quindi chiesto il nome da dare all'oggetto; successivamente, un secondo menu s'informerà di quale sia la struttura desiderata per i rilievi, e noi selezioneremo l'opzione N. 4. Finalmente il programma ci chiederà le dimensioni dell'oggetto, l'altezza del rilievo ed il relativo colore. L'EGG produrrà questo primo oggetto in svolgimento orizzontale, per cui dovremo utilizzarlo come faccia superiore relativa del nostro cubo. Ripetendo l'operazione precedente (con gli stessi termini) otterremo un nuovo oggetto i cui rilievi saranno differenti dal precedente dato che, come abbiamo visto, il parametro inerente ai rilievi viene preso in considerazione come livello massimo d'altezza, e che i punti intermedi vengono generati tramite un algoritmo pseudo random (quasi casuale). Poiché un cubo è composto da sei facce, sei saranno gli oggetti che dovremo generare. La fase successiva passa tramite l'OCT o direttamente dal modellatore dello Sculpt 3D (entrambi permettendo di operare per rotazione e traslazione) dato che dobbiamo ruotare e traslare i sei oggetti in modo da posizionarli adeguatamente

per ottenere un cubo. Nel caso si intenda utilizzare l'OCT per effettuare l'operazione, la prassi da seguire è la seguente: caricato il programma, verrà chiesto il nome dell'oggetto da modificare (oggetto che deve obbligatoriamente trovarsi nella directory GEO) e, successivamente, se si intende modificarlo. Innanzitutto dovremo dare le nuove dimensioni e, poiché non intendiamo variarle, dichiareremo tre 1 (X Y Z). Quindi sarà chiesta l'eventuale variazione di posizione, alla quale risponderemo 0 0 0 e, finalmente, l'eventuale rotazione attorno ai tre assi (in ordine X, Y e Z). A seconda di ciò che desideriamo ottenere, un lato o l'altro, faremo ruotare l'oggetto di 90 gradi, oppure di 180 nel caso si tratti della faccia inferiore del cubo. Volendo invece utilizzare direttamente il modellatore dello Sculpt 3D, dovremo selezionare un punto, tramite il mouse, della faccia del cubo da spostare e da ruotare dopo di che, per selezionare i restanti punti dell'oggetto, andrà scelta l'opzione dalla MenuBar EDIT/SELECT/CONNECTED (per MenuBar si intende la linea posta in alto sullo schermo sulla quale, solitamente, compare il nome del programma e dalla quale si sviluppano i menu a discesa tipici di Amiga). Per effettuare le operazioni di rotazione, selezionate i gadget posti in alto a sinistra su ogni finestra, tenendo conto che se durante la selezione viene tenuto premuto il tasto Amiga posto a destra la rotazione avverrà a passi di 45 gradi e che detta rotazione avverrà relativamente al punto in cui è stato posto il cursore a forma di stella. Una volta ruotato l'oggetto, i suoi punti vanno deselezionati prima di ripetere l'operazione con un altro oggetto, il tutto tramite l'opzione EDIT/DESELECT/ALL. Questa fase va ripetuta per tutti gli oggetti che debbono essere ruotati (nel nostro caso 5). Ora, per assemblare le sei facce del cubo possiamo operare ancora tramite l'OCT, ma è preferibile farlo con lo Sculpt 3D visto che con esso l'operazione è molto più semplice; basterà selezionare con il mouse un punto della faccia che desideriamo muovere, quindi scegliere dalla MenuBar l'opzione EDIT/SELECT/CONNECTED. Tutta la faccia in causa verrà evidenziata e per posizionarla basterà selezionare il gadget a forma di pinza posto sul lato sinistro delle finestre (questo gadget aggancia l'oggetto al cursore in modo che, muovendo il mouse, si muoverà anche l'oggetto). L'operazione va ripetuta per tutte le facce, non dimenticando di deselezionare l'oggetto spostato prima di selezionarne e spostarne un altro, il tutto con l'opzione EDIT/DESELECT/ALL. A questo punto l'oggetto che desideravamo esiste, e può finalmente essere «processato» nelle strutture che seguono, ma può anche venir rimanipolato dai modellatori per ottenere variazioni successive. Ad esempio: come sappiamo, la struttura dei frattali con i quali abbiamo costruito il nostro oggetto è tale da presentare tagli netti tra un poligono e quello adiacente quindi, volendo ottenere un risultato più verosimile, dovremo rimodellare l'oggetto tramite la funzione EDIT/DO/SUBDIVIDE dello Sculpt 3D, la quale raddoppierà i punti costituenti la superficie del nostro oggetto. Dovremo successivamente operare, tramite l'opzione EDIT/SMOOTHING, in fase di rendering, per addolcire le curve.

■

Software Express

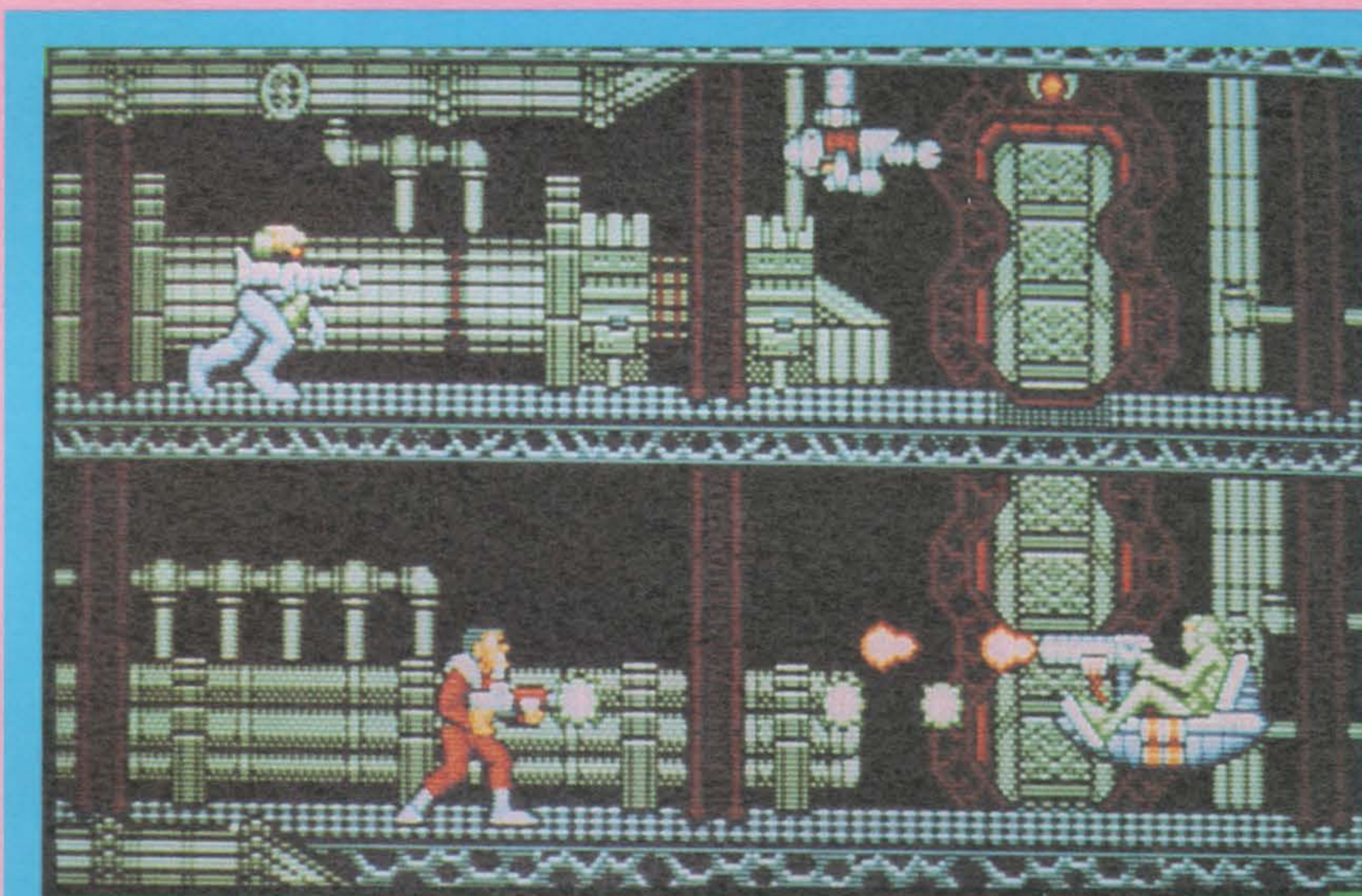
a cura di GIULIO BONIFAZI

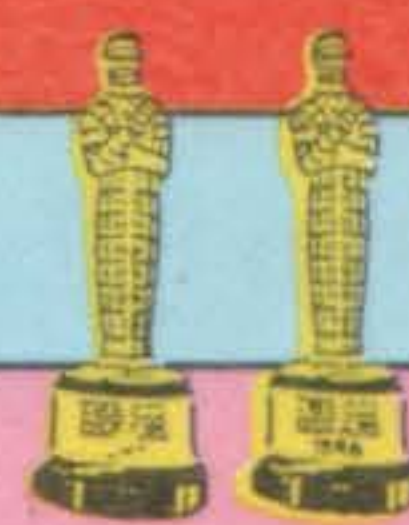
Brevi recensioni di giochi ed utility freschi di giornata. Visti, giocati e valutati per voi.

SUFFICIENTE DISCRETO BUONO OTTIMO!

OBLITERATOR

PSYGNOSIS LTD.

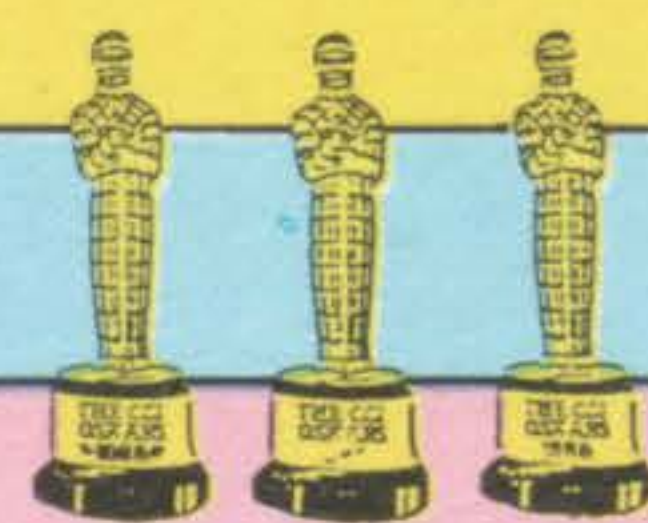
L'ottima presentazione lascia presagire la qualità del gioco, tra i migliori in circolazione soprattutto dal punto di vista grafico e per l'accuratezza degli sprites. La storiella è piuttosto semplice e banale: impersonerete Drak, l'ultimo degli obliterators, l'ultimo guerriero. La povera Terra è sottoposta all'attacco degli alieni e la vostra missione consiste nell'intrufolarvi nella nave nemica per metterla fuori uso. Ovviamente i vostri avversari tenteranno di fermarvi in tutte le maniere, ma ce la dovrete fare, ne va della salvezza della Terra! Lo scopo vero e proprio del gioco è quello di trovare cinque oggetti fondamentali che rappresentano i motori dell'astronave, le barriere difensive nemiche, le armi aliene, il computer principale ed una sorta di chiave a forma di rombo, indispensabile per fuggire al momento opportuno. Una volta recuperato tutto, dovrete uccidere il maggior numero possibile di nemici per guadagnare punti, poiché dopo pochissimo tempo il vostro punteggio inizierà a calare ed assumerà le funzioni di un vero e proprio conto alla rovescia. Arrivato a zero, tutti i guerrieri nemici vi attaccheranno e se a questo punto non sarete già riusciti ad arrivare alla vostra navetta, sarete distrutti. Obliterator non è un classico shoot em up (spara e fuggi), ma appartiene al nuovo filone degli arcade adventure: non si tratta certo di un'avventura, si badi bene, ma il personaggio è controllato tramite un pannello nel quale sono raffigurati tutti i simboli necessari con delle icone che rappresentano tutte le azioni possibili quali sparare, muoversi, entrare, difendersi, agire. Tanti i tipi di arma da trovare, dalla pistola al bazooka. Un particolare non trascurabile è costituito dalle munizioni sparse un po' dappertutto, che DEVONO necessariamente essere raccolte per aumentare l'autonomia dell'arma attualmente selezionata. Ogni arma ne avrà certamente bisogno, poiché sono previsti differenti tipi di icona per le diverse munizioni.

GLI ANTENATI

HANNA & BARBERA

Altra versione Amiga di una notissima serie di cartoni animati, protagonisti Barney, Fred e le rispettive gentili mogliettine Betty e Wilma. Il copyright H&B garantisce la fedeltà dei disegni dei personaggi, perfettamente uguali a quelli dei cartoons originali. Tutto è ricostruito nei minimi particolari, anche l'espressione non proprio felice di Barney che deve sempre sottostare agli ordini della moglie. Questa volta il lavoro affibbiatogli è davvero duro: verniciare tutta la palizzata che circonda la casa, tenendo a bada anche la pestifera figlioletta, che ovviamente non ne vuol sapere di starsene buona. Detto così non sembrerebbe esserci niente di speciale o di difficile da fare, segno evidente che non avete acquistato il gioco e non avete ancora perso mezza giornata per verniciare due centimetri di muro! A parte lo stress per le difficoltà che incontrerete, smanettando il tempo passerà senza accorgervene e sarà facile che restiate incastrati con il joystick in mano a lungo. Certo viene voglia di gettare la bambina dal buco della palizzata quando, dopo aver fatto tanta fatica per ripulire tutto, la carognetta esce dal box ed imbratta il muro con i suoi stupidi disegni. Dovrete allora lasciare il pennello (che in realtà è un povero pappagallo), riportare la piccola peste nel suo box, riprendere il pennello volante (!), passare vicino al secchio di vernice e continuare il lavoro. Se non ne avete abbastanza, sappiate che dovrete spostare in continuazione la scala per accedere alla parte superiore della palizzata e che avrete un tempo massimo entro il quale terminare l'opera: altrimenti sentirete Wilma, quando ritorna!

THE THREE STOOGES

CINEMAWARE

Clamoroso gioco zeppo di fantastiche immagini; la presentazione è semplicemente eccezionale, anche se ormai siamo abituati alle trovate di questa software house americana. Stavolta sembra proprio l'inizio di un film, con tanto di caratteristico conto alla rovescia che fa bip bip, mentre le scritte appaiono sui cartelloni pubblicitari che costeggiano un sentiero. I tre simpaticissimi protagonisti esordiscono facendo strani rumori (che non definiremo per... pudore) sulla famosa schermata originale di «Defender of the crown», per poi proseguire il loro cammino ed arrivare di fronte ad una casetta, dalla quale sembrano uscire decine di bambini. I nostri eroi si fermano e gentilmente chiedono cosa stia succedendo alla signora, presumibilmente la mamma della nidiata di ragazzini: tra le lacrime la donna lamenta di non avere una lira per mettere a posto la casa, se non trova i soldi necessari entro trenta giorni perderà tutto. «Niente paura, ci pensiamo noi!» è la fatidica frase pronunciata cavallerescamente sì, ma forse un po' troppo in fretta. Inizia così la storia rocambolesca dei tre personaggi: dovranno fare i salti mortali (cioè, glieli dovrete far fare voi) per riuscire a procurarsi i dollari necessari, sfiancandosi in qualsiasi tipo di lavoro. Molto divertente l'assunzione nel ristorante per il lancio delle torte, ma neppure l'incontro di boxe è da perdersi! I tre dovranno perfino prendersi a cazzotti tra loro; fare a botte per guardare attraverso la fessura di una palizzata; ricevere i consigli, non molto salutari, di un vecchio boss e farne e passarne di tutti i colori. Voi controllate con il joystick il personaggio di turno attualmente impegnato nel lavoro; quando si tratterà di scegliere la strada da prendere, sarà necessaria una grande prontezza di riflessi per non finire sempre negli stessi punti, o per evitare di raccogliere i mucchietti di soldi già intascati precedentemente. I quattro Oscar risultano addirittura pochi visto il valore di questo gioco, senza dubbio uno dei migliori in circolazione per Amiga fino a questo momento. Avventurieri e no, dovrete tutti passare per i Three Stooges!

KEYBOARD CONTROLLER SEQUENCER

EMILE TOBENFELD

Nuovissimo programma musicale, dedicato esclusivamente a tutti i possessori di Midi e relativi strumenti compatibili. Numerosissime le opzioni, tutte molto sofisticate: dalla semplice selezione dei canali midi, al rilevamento del pedale e della sincronizzazione esterna. Si tratta di un sequencer step by step multitraccia, che può memorizzare ben 36 piste differenti, ognuna totalmente indipendente dalle altre. Per risparmiare memoria non è disponibile il Workbench, con i suoi cari menu a discesa e relativa selezione. Inizialmente l'impatto non è dei migliori ma, poiché questo potente programma è dedicato ad esperti musicisti Midi-dipendenti, il danno non è grave! Il cursore a forma di mirino semplifica comunque notevolmente tutte le operazioni, anche se alcune di queste sono ottenibili direttamente premendo i tasti funzione. Nel dischetto sono presenti due demo, una delle quali contiene ben 8 tracce piene di musica. L'altra ne contiene soltanto una, molto più banale. Durante l'esecuzione di un brano è possibile selezionare tutte le tracce oppure metterne qualcuna in pausa, oppure ancora registrare contemporaneamente su di un'altra. Per parlare come si deve di questo Editor Midi avremmo bisogno di tutto lo spazio dedicato al software; non essendo possibile, rimandiamo il tutto ad una prova vera e propria con batteria e tastiere.

TEXT PRO

DATA BECKER

Un semplice ma molto potente word processor che si va ad aggiungere alla numerosa famiglia dei programmi di trattamento testi. È possibile caricare disegni in formato IFF che possono essere convertiti in bianco e nero, oppure lasciar decidere al programma stesso la palette dei colori. Il testo può circondare il disegno oppure proseguire nelle righe successive, anche se il risultato non si può certo paragonare a quello ottenibile con altri word processor, molto più sofisticati di questo. La potenza del Text Pro risiede nella sua estrema facilità d'uso, che lo riserva a chi si avvicina per la prima volta a simili utility. Tutte le opzioni si ottengono dalla menu bar, le più usate possono aversi anche con la combinazione del tasto Amiga. Costituiscono due simpatiche ciliegine la Date ed il Time: clickandoci sopra, compaiono automaticamente nel testo la data o l'ora attuale, utile quest'ultima per intestare, ad esempio, le lettere. Notevole la possibilità di trasferire file o di riceverne attraverso la porta seriale, inizialmente settata a 9600 baud, 8N1. Esiste inoltre l'opzione di scelta del formato del testo; se mantenere cioè quello caratteristico del programma, oppure se adottare quello normale ascii. Si potrà così usare il Text Pro anche come editor di sorgenti per i vari compilatori. Chiare le intenzioni dei produttori di dare il massimo nel modo più semplice, ma non sappiamo fino a che punto siano riusciti nell'intento, tenendo conto anche del prezzo.

POOL

MASTERTRONIC

Conversione elettronica del biliardo o, più propriamente, della carambola. È naturalmente possibile giocare in due oppure sfidare il computer, che si rivela un avversario molto preciso, anche se ogni tanto prende delle vere e proprie cantonate. La dinamica del gioco è molto semplice: con il mouse si decide la direzione da dare alla boccia bianca, quella che deve essere colpita con la stecca; poi, entro un tempo limite che appare sulla boccia mirino, si passerà ad un secondo quadro. Qui dovrete decidere tutto quanto compete ad un tiro professionale, dalla forza all'angolo di tiro, dal topspin al backspin. Ovvero deciderete se volete che la vostra boccia tenda a proseguire la sua corsa dopo l'impatto (topspin), oppure tenda a fermarsi e, al limite, a tornare indietro (backspin). Anche qui avete un limite massimo di tempo, scaduto il quale scaglierete il tiro con i valori che in quel momento sono segnati. Non è possibile scegliere numericamente la potenza del tiro, si deve attendere il momento opportuno, poiché la barra si muove continuamente dal minimo al massimo. È necessario stare molto attenti nel scegliere le bocce da mandare in buca, altrimenti il programma sintetizzerà un bel «Foul shot» (tiro sbagliato) anche se avrete centrato la buca.

CRACK

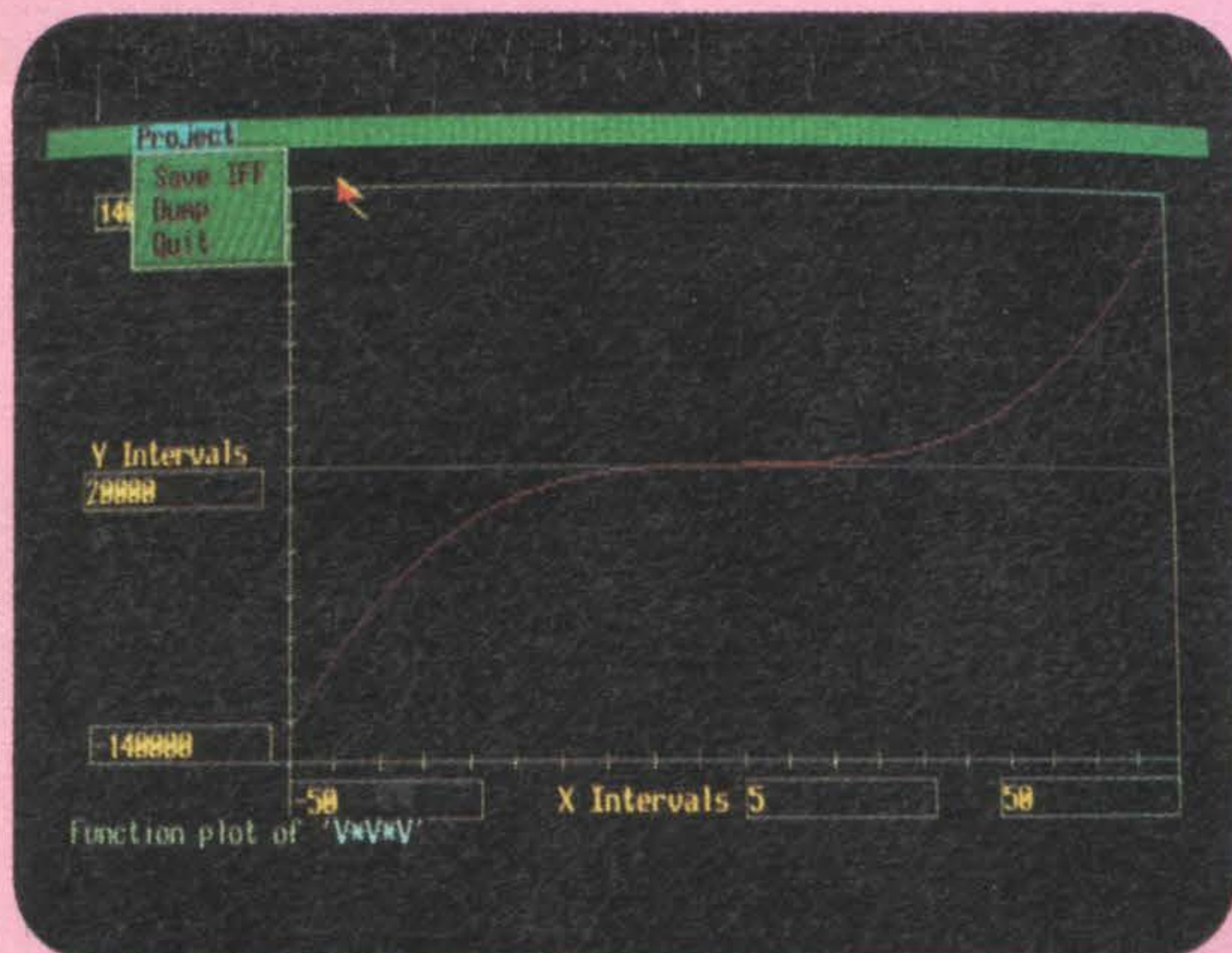
LINEL

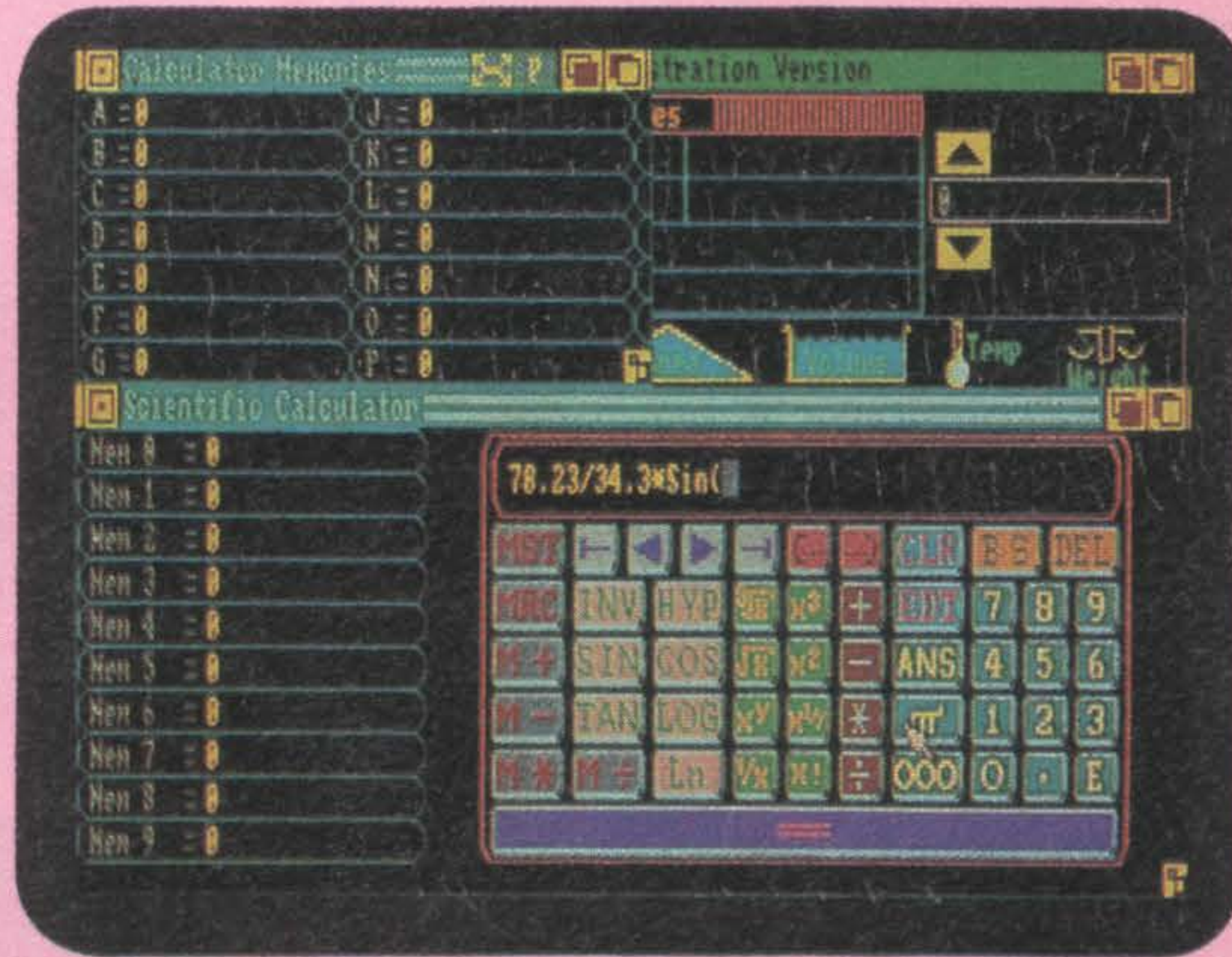
Ulteriore versione dell'immortale Breakout. Personaggi principali: un muro di mattoncini colorati, una racchetta rettangolare ed una pallina distruttrice. Se pensavate che con «Arkanoid» questo genere di gioco fosse stato archiviato beh, avete preso un grosso granchio. In teoria non si dovrebbe sostenere che Crack sia un gioco originale, visto il filone cui appartiene; noi invece sosteniamo questa tesi a pieno merito. Tutto il gioco è ambientato nella preistoria e la sola presentazione meriterebbe una recensione a parte: l'urlo dell'antenato di Tarzan ed il suo tonfo ricordano un po' Fantozzi nei suoi film. Naturalmente anche il set di caratteri, appositamente disegnato, rende bene il concetto che siamo qui nella preistoria. Una delle novità di questo gioco è la presenza di un potentissimo e semplicissimo editor per costruire schermi a proprio piacimento, potendo inserire tutto l'immaginabile: dalla scelta del colore del mattoncino al numero di colpi necessari per abbatterlo, ai bonus da introdurre nei punti ritenuti strategici. A fine lavoro si può salvare il nuovo schermo o provarlo direttamente. Inoltre è possibile giocare contemporaneamente in due con l'opzione Battle Mode.

ENFORCER

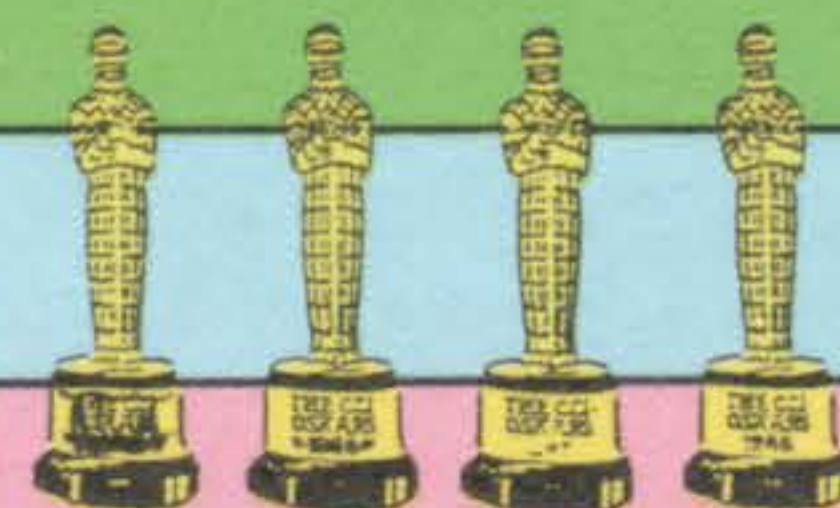
METHODIC SOLUTIONS

Simpaticissimo game ispirato al famoso e veloce Prohibition, anche se molto più ricco di azione e con diverse variazioni. Probabilmente gli autori si sono riferiti alla fortunata serie di film «Scuola di Polizia»; tutto il gioco è infatti visto come una serie di lezioni impartite ad un poliziotto novello per il quale è molto semplice combinare ogni tipo di guai! Già il primo impatto con il livello numero uno si può rivelare una catastrofe: apparentemente molto semplice, in realtà si dimostra davvero arduo dover assemblare i pezzi di varie pistole, messe in ordine sparso. Dall'alto pioveranno un'infinità di colpi che elimineranno tutti gli oggetti che riusciranno a colpire. Spostando a destra ed a sinistra tutti i componenti, ne riuscirete magari a salvare qualcuno, ma la velocità è tale che probabilmente vi rimarranno solo dei pezzi non compatibili tra loro perché appartenenti a modelli di pistole diversi! Comunque, qualora anche non riusciate a combinare niente, passerete alla seconda lezione, molto più divertente. In questa nuova fase muovete il mirino della vostra arma con il joystick cercando di eliminare tutti i possibili malviventi che circolano per la strada. Sagome raffiguranti bambini, signore, arabi, punk, poliziotti e tanti altri personaggi si alterneranno velocemente sul marciapiede, fermandosi per alcuni brevissimi istanti. In queste frazioni di secondo è necessario decidere se sparare o meno, il che equivale ad attribuire pericolosità o meno al personaggio, senza contare che non sarà sufficiente neppure prendere una buona mira perché il più delle volte i protagonisti da eliminare saranno sovrapposti (lasciando libera soltanto una piccola parte) alle sagome degli innocenti. Inoltre, nella parte alta dello schermo passa regolarmente, o quasi, un classico bersaglio che regala punti se centrato. Non è un gioco complicato, affatto, ma senza dubbio mette a dura prova la prontezza di riflessi di chiunque. Il bello è che, sovente, alcuni cattivi non si fermano neppure per un istante, ma scorrono solamente; in questi casi colpirli è davvero arduo, perché ci saranno sempre altre persone tra voi ed il bersaglio. Uno dei trucchetti più usati per farvi cadere in fallo è quello di disegnare per due volte nello stesso punto la sagoma di un bandito e, appena un secondo dopo, quella di una donna. Sulle prime farete, c'è da scommetterci, una strage di innocenti! Giochino molto divertente e veloce, che rischia di tenervi ipnotizzati per ore davanti al secondo livello nella vana speranza di passare al terzo (come è accaduto a chi scrive queste righe! Ma, vi prego, non ditelo a nessuno...). ■

MATH AMATION

PROGRESSIVE PERIPHERALS

Interessantissima utility che potrebbe essere, ma solo a prima vista, paragonata ad una sofisticata calcolatrice scientifica. In realtà questa similitudine le sta molto stretta, poiché le sue funzioni vanno ben oltre quelle di una calcolatrice, seppur complessa. Tra le peculiarità del programma, la possibilità di eseguire calcoli statistici, algebrici e geometrici; ben 36 memorie generali; la presenza del sistema WIMP, che altro non è che il sistema di finestre e menu a discesa caratteristico di Amiga, ed un potente editor di espressioni. Per gli smemorati ad oltranza ricordiamo che questo Math Animation riesce a tener memorizzati gli ultimi 50 risultati, quali che siano! Entrambe le calcolatrici scientifiche e matriciali sono completamente programmabili: la vostra espressione potrà essere lunga fino a 200 caratteri; se siete proprio dei geni, addirittura fino a 600. In pochissimi secondi viene tracciato il grafico di una funzione di due variabili con l'intervallo definibile, del quale viene calcolato immediatamente l'integrale, ovviamente definito nello stesso intervallo di definizione. Tra le opzioni di save/load, molto importante quella che salva in formato IFF il grafico ottenuto, che potrà quindi essere rielaborato con i relativi programmi grafici. Il modulo riservato al business permette di visualizzare tabelle di dati nei consueti grafici a torta ed a barre verticali od orizzontali in maniera semplicissima. È inoltre immediato il convertire misure di ogni tipo da un'unità ad un'altra. Queste sono solo alcune delle diverse opzioni offerte dal programma.

SIDEWINDER

ARCADIA

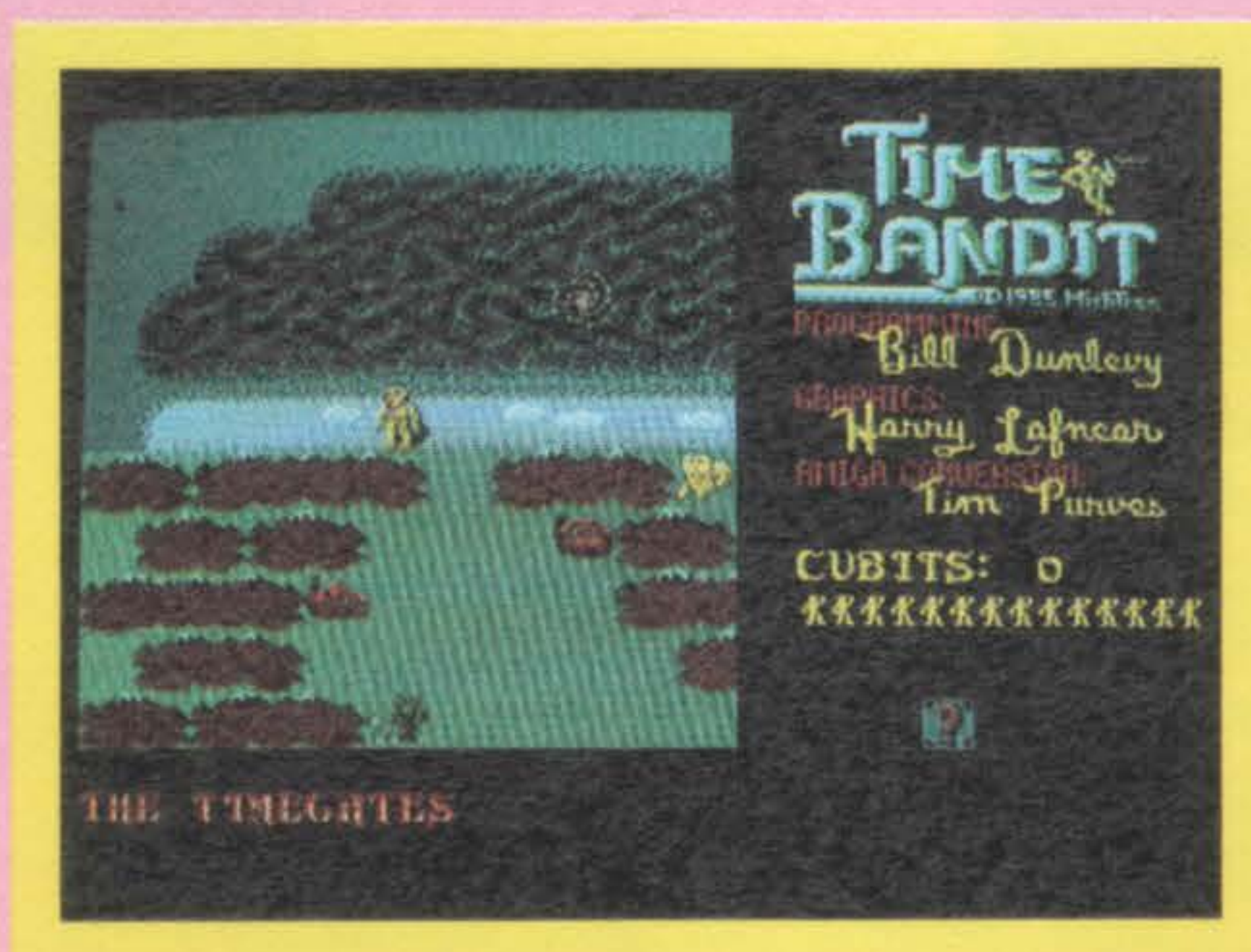
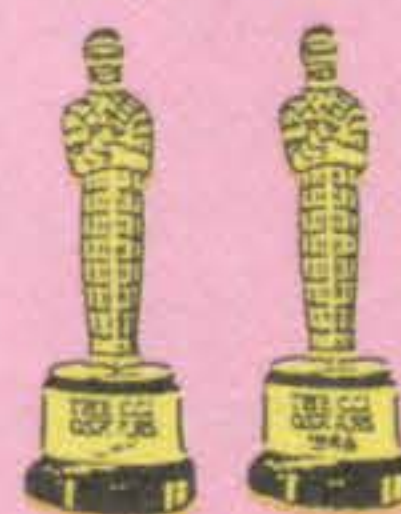
Velocissimo shoot em up, naturale antagonista dei suoi predecessori «XR35» e «Xenon». La pulizia dei movimenti e la scelta dei colori sono decisamente ottime, il controllo dell'astronave perfetto anche nei livelli più difficili. Inutile raccontare la missione da portare a termine perché avete già capito che si tratta di un gioco nel quale vale la famosa regola «Spara a tutto quello che si muove» che, se applicata alla lettera, non ammette errori, visto che non ci sono navicelle alleate o costruzioni da salvaguardare; tutto vi è ostile, anche quelle cose molto semplici di forma quadrata che apparentemente non lanciano nulla: provate a passare nelle loro vicinanze e ve ne accorgerete! Gli sprite arrivano velocissimi e scompariranno altrettanto rapidamente se non li farete fuori. Nel primo livello i problemi si affacciano soltanto alla fine perché è abbastanza semplice, sulle prime, evitare le ondate nemiche e le loro bombe; quando però tutto sembra finito e la pedana di atterraggio è ormai vicina, ecco due navi prendono a seguire costantemente i vostri spostamenti, due veri e propri kamikaze. Gli smanettoni capiranno subito come evitarle, ma per gli altri non sarà uno scherzo. Non vi lamentate già da adesso però, perché almeno all'inizio del gioco riuscite a vedere da dove partono le mine e ad eliminare i cannoni con una raffica di colpi. Dal secondo quadro in poi, invece, neanche per sogno! Sbucheranno dal terreno delle torrette che, appena lanciate le micidiali bombe, si ritireranno immediatamente. Dei simpatici robottini vi attendono impazienti nel giardino, sempre se riuscite a passare il deserto iniziale... State ancora leggendo queste righe? Ancora non avete resettato per caricare Sidewinder? Male, molto male...

CAPONE

ACTION WARE

Ovviamente ispirato alla mafia ed ai suoi loschi affari, in questo arcade - adventure le sparatorie la fanno di gran lunga da padrone. Apparentemente sembra tutto tranquillo nella cittadina teatro del gioco, che ha tutta l'aria di essere una ricostruzione di uno di quei centri caratteristici del Far West, ma dovrete stare attenti ad ogni piccolo particolare perché le insidie sono nascoste dappertutto. Dietro ogni finestra c'è un fucile pronto a far fuoco e sarà molto difficile evitare i colpi nemici. La software house produttrice di Capone mette in vendita, insieme al gioco, una pistola ottica che dovrebbe facilitare molto l'avventura. Noi l'abbiamo provata con il solito joystick, e già ci è sembrato sufficiente, anche se, logicamente, sparare con la pistola sarà certo un'altra cosa e darà indubbiamente quel tocco in più di realismo e brivido che manca nel gioco puro e semplice. La grafica è un po' solita, caratterizzata da sprite piuttosto simili. Il gioco è, di per sé, divertente e la presenza di diversi quadri, sufficientemente differenti tra loro, spinge il giocatore a proseguire la missione, anche se è affatto facile. Rubare i soldi dalla cassaforte della banca sarà davvero un'impresa, visto il numero di candelotti di dinamite che piovono da tutte le parti. Quando arrivano questi simpatici regalini, c'è poco da fare: lo schermo esplode in una sequenza multicolore molto appariscente e vi lasciamo immaginare che fine avrete fatta voi!

TIME BANDIT

MICRODEAL

Arcade adventure, questa, molto sofisticata, caratterizzata dal quadro di gioco leggermente rimpicciolito rispetto al solito. Anche in questo caso non si tocca la tastiera per giocare, se non in casi eccezionali, ad esempio quando ci si trova davanti ad una porta chiusa. Magicamente appare la richiesta di un codice segreto senza il quale è impossibile entrare e che naturalmente non è facile da scovare; molto più semplice impadronirsi di una chiave, facilmente visibile, che viene immediatamente visualizzata nella parte bassa dello schermo. Il personaggio che controllate è molto piccolo rispetto allo scenario che lo circonda: questo soprattutto all'inizio, quando tutto quello che vi circonda rappresenta una sorta di paesaggio in miniatura: basta infatti avvicinarsi ad una qualsiasi di quelle piccole casette ed accennare ad entrare e tutto cambia. La mappa del gioco è molto vasta, a giudicare da quella piccola parte che siamo riusciti a perlustrare. Rilevante la possibilità di giocare contemporaneamente in due; in questo caso compaiono due visuali dei rispettivi personaggi e, se l'altro si vuole divertire alle vostre spalle, può inseguirivi e spararvi addosso. Se non fuggirete in tempo vedrete apparire, al posto della solita sagoma, l'ombra di quello che eravate fino a qualche minuto prima! Un simpatico messaggio vi avvertirà della trasformazione, ma quando non ci sarà più nulla da fare. ■

THE PINK PANTHER

RELINE

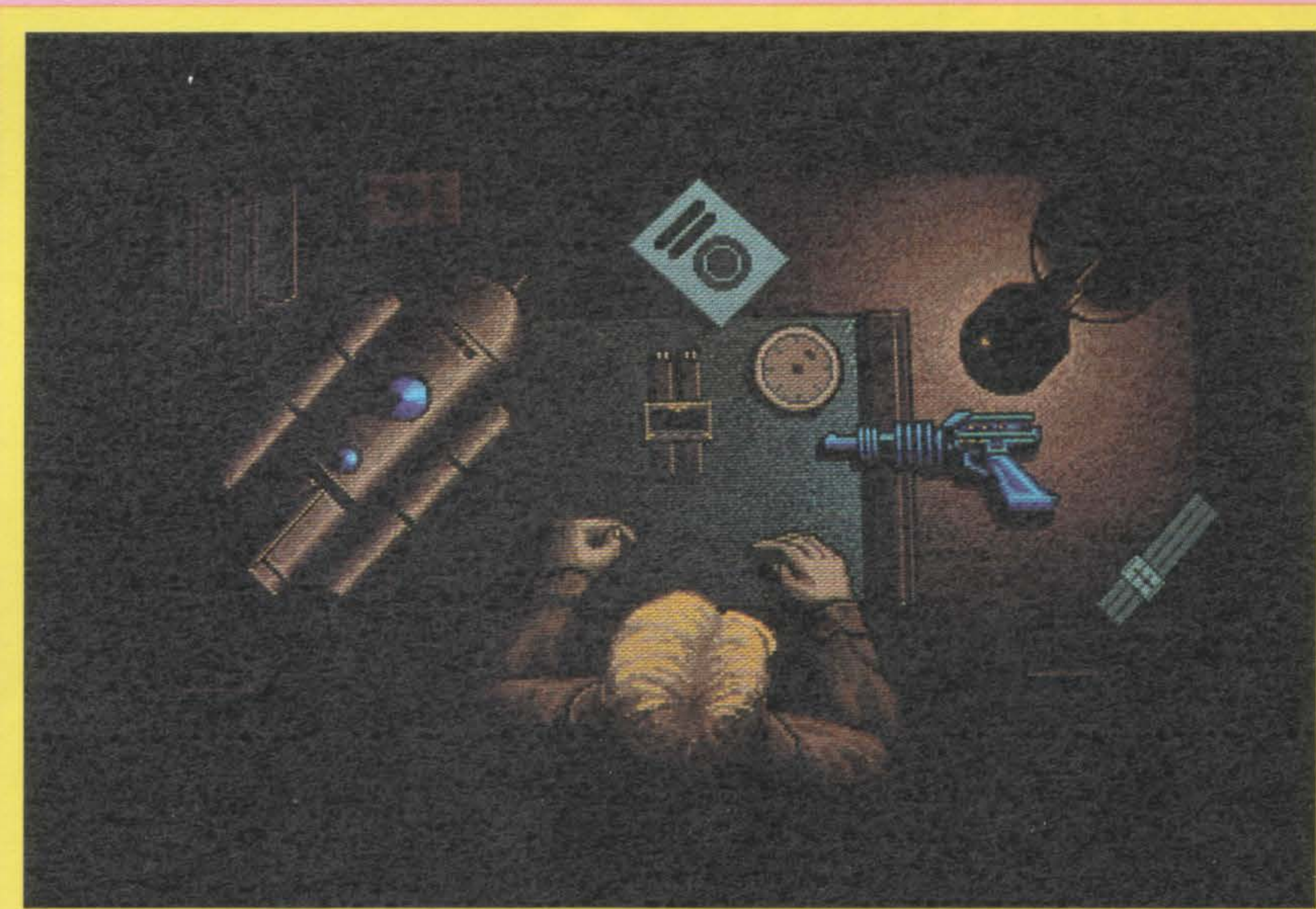
Versione computerizzata delle celeberrime avventure con protagonisti la Pantera Rosa ed il suo ineffabile nemico, l'ispettore Closeau. La nota musichetta accompagna tutta la prima parte del gioco, sempre pronta, anche in seguito, a suonare le sue note! Dovrete guidare la tremenda Pantera Rosa in una nuova puntata dell'eterna sfida con l'ispettore, non solo comandando gli spostamenti, ma interagendo nel gioco, scegliendo per esempio gli oggetti da chiedere nel negozio e svariate cose di questo tipo. Tutto si controlla con il joystick: anche questo gioco, infatti, è un ibrido, un po' arcade un po' avventura, ma veramente simpatico. Una delle tante scene spassose è quella in cui la Pantera, grazie ai nostri comandi, arriva in una stanza nella quale il tremendo ispettore sta passeggiando in pigiama, in preda a... sonnambulismo! Dovete continuare il vostro lavoro (non vi diciamo certo quale!) senza disturbare nessuno e senza far cadere nulla, altrimenti Closeau si sveglia e vi porta via ammanettati! Il supermarket è fondamentale per accedere ai vari livelli, poiché dovrete sempre avere con voi un particolare oggetto che vi potete procurare soltanto lì, non senza aver faticato un pochino. È anche previsto un bel riposino per la svogliata Pantera Rosa, naturalmente se e quando avrà portato a termine tutti i lavori. Una grafica non eccellente ed il movimento degli sprite non proprio finissimo renderebbero questo gioco appena sufficiente, ma la simpatia dei personaggi e delle varie scenette che caratterizzano la storia lo salvano rendendolo davvero carino. ■

ROCKET RANGER

CINEMAWARE

Eccellente lavoro che soddisfa le aspettative dei fans delle avventure di questa Casa, Rocket Ranger era molto atteso vista la pubblicità americana che ne annunciava l'arrivo da parecchio tempo. La storia è ambientata nella seconda guerra mondiale, e tutte le schermate sono curate sin nei minimi particolari. In alcuni casi è stato scelto il bianco e nero al posto della miriade di colori, e questo la dice lunga sullo studio a tavolino eseguito dai programmatori. La presentazione è davvero ben fatta, degna di un computer grafico professionale, anche se non si rileva spreco di effetti speciali. Ci sono disegni che fanno addirittura impressione per la precisione, la scelta della prospettiva e quella del colore, soprattutto quello raffigurante una tortura nazista. Il gioco provato da noi è soltanto una beta version, non ancora definitiva, quindi non possiamo esprimere più severi ed argomentati giudizi; appena in possesso della versione definitiva la spulceremo ben bene per verificare in maniera più accurata tutto il gioco anche se, da quello che abbiamo visto, ci sentiamo già di assegnargli quattro Oscar.

CONSIGLI E SOLUZIONE DI

THE LAND OF THE LOUNGE LIZARDS

Avventurati con noi nel dedalo spesso difficilmente districabile delle adventure. Se sei uno di quei pochi in gamba che son riusciti a risolverne, inviaci la prova della tua bravura; soluzioni e mappe verranno pubblicate in questo spazio dedicato ed il tuo nome citato.

a cura di MARCO BROVELLI

Gli ideatori di adventure games non brillano ahinoi sovente per originalità nell'ambientare le loro creazioni: in quante dozzine di giochi ci sono cavalieri che devono salvare una principessa rapita o recuperare un tesoro?

Quante volte, in un'avventura, avete rivestito i panni dell'intrepido esploratore nella giungla o nel sottosuolo? Per non parlare dei giochi di ambientazione «fantasy», ispirati più o meno direttamente alla saga de «Il signore degli anelli»: elfi, gnomi e follette ormai non si contano più nella miriade di avventure loro dedicate.

Dall'America è arrivata invece una ventata di novità: «Leisure suit Larry in the land of the lounge lizards» è un'avventura molto realistica ed innovativa, in cui dovrete affrontare situazioni tipiche della vita, a volte spiacevoli ma spesso anche eccitanti. Talmente eccitanti che la Sierra On-Line, la software house che l'ha ideata e prodotta negli Stati Uniti, ne sconsiglia l'acquisto ai minori di diciotto anni! Non ci sono immagini pornografiche o scabrose, ma il sesso riveste un ruolo non indifferente nella vicenda, visto che lo scopo del gioco consiste nel trovare una ragazza, conquistarla, sposarla e trascorrere piacevolmente con lei la prima notte di nozze...

Prima di passare alla soluzione del gioco, alcuni consigli: esaminate tutto quello che trovate, perché, anche se non sempre ciò è utile ai fini del gioco, le risposte sono spesso divertenti. I movimenti del protagonista sono comandati dal joystick, quindi attenti a dove mettete i piedi: è facile finire sotto un'auto se camminate in mezzo alla strada, o essere picchiati e rapinati se vi avventurate in vicoli bui e andate a sbattere contro persone dall'aria poco raccomandabile.

All'inizio del gioco vi trovate davanti ad un bar. Entrate, andate al banco, sedetevi e ordinate uno whisky; non bevetelo, alzatevi e passate per la porta in alto a sinistra, oltre la quale troverete un ubriaco a cui darlo; in cambio otterrete un telecomando. Esaminate il tavolo e prendete la rosa, quindi entrate nel bagno; dopo esservi seduti sul water per le «faccende di rito», andate vicino alla parete e leggete i graffiti incisi; esaminate il lavandino e prendete l'anello. Tornate nel bar, andate alla porta chiusa in basso a destra, bussate e dite la parola d'ordine che avete letto sul muro («Ken sent me»); entrate ed accendete la televisione con il telecomando: continuate a cambiare canale fino a quando non compare un film porno sullo schermo ed attendete fino a quando il buttafuori non si ferma a guardare interessato. A questo punto salite le scale, entrate nella stanza e andate al tavolo sul quale, esaminatolo, troverete una

ADVENTURE

scatola di dolci. Prendetela ed aprite la finestra; arrampicatevi sulla scala anti-incendio, scendete e lasciatevi cadere nel bidone delle immondizie; prendete il martello che vi è contenuto ed arrampicatevi fuori poi, camminando, andate a sinistra davanti all'ingresso del bar. Chiamate un taxi, salitevi e chiedete al conducente di portarvi al Casinò; una volta arrivati, pagate ed uscite dalla vettura. Se appare un uomo con indosso un barile, comprategli una mela, quindi entrate nel locale.

Salvate la situazione e puntate il massimo giocando a Blackjack o con le Slot-machines: se vincete, salvate di nuovo; se perdete, ricaricate la situazione precedente, continuando a giocare fino a quando non avrete vinto almeno 300 dollari. Lasciate ora la sala da gioco passando per l'uscita in alto; esaminate il portacenere e prendete la tessera, poi entrate nel cabaret e sedetevi sulla sedia in basso a destra.

Dopo lo spettacolo, uscite dal casinò, chiamate un taxi e fatevi portare in discoteca. Per entrare dovete mostrare la tessera al buttafuori. All'interno, andate al tavolo al quale è seduta la ragazza ed accomodatevi accanto a lei. Guardatela attentamente (ne vale la pena!), parlate con lei e chiedetele di ballare; dopo esservi esibiti, tornate al tavolo e datele la rosa, i dolci e l'anello. Quando vi chiederà del denaro datele 100 dollari, uscite e chiamate un taxi. Fatevi portare davanti alla cappella nuziale, dove incontrete uno strano individuo: parlate con lui, entrate nella chiesa ed avvicinatevi alla ragazza; pagate il sacerdote e, dopo il matrimonio, uscite ed andate a piedi fino al casinò, a sinistra. Entrate, arrivate all'ascensore e salite al quarto piano; avvicinatevi alla porta contrassegnata dal cuore dipinto e bussate; accendete la radio ed ascoltate attentamente la pubblicità. Uscite dalla stanza e dal casinò, chiamate un taxi e fatevi portare al negozio aperto tutta la notte. Esaminate il telefono pubblico e leggete il numero scritto sul lato, chiamatelo e rispondete alle domande che vi verranno poste; poi telefonate alla Sierra On-Line, il cui numero è segnato sul biglietto nel vostro portafogli. Entrate nel negozio, andate davanti allo scaffale dei giornali, prendete una rivista e leggetela; avvicinatevi alla parete in fondo al locale, prendete una bottiglia di vino e dirigetevi al banco; chiedete un preservativo al commesso, pagate ed uscite. Quando squilla il telefono, rispondete e date la bottiglia di vino al mendicante fuori dal negozio, prendetegli il coltello e chiamate con il telefono il numero sentito alla radio in precedenza (555-8039). Fatevi portare una bottiglia di vino in albergo, al «Casino hotel honeymoon suite», poi chiamate un taxi e tornate al casinò. Andate nella camera d'albergo al quarto piano, entrate ed avvicinatevi alla ragazza; versate il vino, infilatevi nel letto e... sorpresa! Dopo che la ragazza se ne sarà andata, tagliate la corda con il coltello e prendetela; quindi scendete e giocate al casinò fino a che non avrete almeno 50 dollari. Usciti dal casinò, tornate al bar ed entrate nella stanza bussando ed usando la parola d'ordine: il buttafuori sarà ancora davanti alla TV, immerso nella visione del film. Salite le scale, avvicinatevi al letto, spogliatevi, indossate il profilattico, entrate nel letto con la ragazza e, dopo aver fatto ciò che è naturale in questi casi, ricordatevi di toglierlo! Rivestitevi, arrampicatevi sulla scala anti-incendio, legatevi un capo della corda attorno al polso ed assicurate l'altro alla ringhiera. Prendete le pillole, rompete la finestra con il martello, tornate sulla scala anti-incendio e slegate la corda dal polso; poi lasciatevi cadere dalla scala nella spazzatura, uscitene e tornate davanti al bar, dove prendete un taxi per tornare al casinò.

Entrate, prendete l'ascensore e salite all'ottavo piano: andate davanti alla guardia, esaminatela e datele le pillole; appena hanno fatto effetto, esaminate il suo tavolo e premete il pulsante. Entrate nell'ascensore privato e salite. Uscite dall'ascensore, camminate dietro al muro in fondo ed entrate nella camera da letto e nell'armadio. Gonfiate la bambola ed usatela (ehm...), poi seguitela fino alla vasca da bagno. Arrivati sul bordo spogliatevi, entratevi e, dopo aver dato la mela alla ragazza, divertitevi: ve lo meritate!

KNIGHT ORC
(Level 9)

Le differenze più eclatanti tra questa ed altre avventure grafiche attualmente disponibili per Amiga non riguardano tanto la trama o i personaggi, quanto piuttosto il metodo di gioco. Non c'è più bisogno di disegnare una mappa per orientarsi, né di annotare la posizione di ogni oggetto trovato per poterlo recuperare più tardi. Questi procedimenti, normalmente indispensabili per la risoluzione di un'avventura, sono resi obsoleti dalla presenza di due nuovi comandi, GOTO e FIND. È sufficiente il comando GOTO seguito dal nome di un luogo per spostarsi da un posto all'altro; e basta digitare FIND seguito dal nome di un oggetto per vedersi trasportati nel luogo dove esso si trova, a patto che non ci siano problemi da risolvere lungo la strada. Altri comandi utili sono: OOPS, che permette di tornare indietro di una mossa nel caso di errori, e RAM SAVE o RAM RESTORE, per salvare o ricaricare la posizione di gioco in memoria invece che sul disco. Non si deve tuttavia credere che grazie a questi comandi risolvere Knight Orc sia uno scherzo; a rendere la vita difficile al giocatore di turno contribuiscono circa cinquanta personaggi, che agiscono indipendentemente e che possono aiutare o intralciare chiunque venga in contatto con loro. Protagonista di Knight Orc è Grindleguts, un orco abbandonato dai suoi compagni, che deve ritrovare la via di casa e tornare nella torre ricostruendo un ponte sospeso. Moltissime schermate grafiche, realizzate con una curiosa tecnica che le rende simili a dipinti ad olio, visualizzano luoghi e situazioni di gioco. Completa la confezione un breve romanzo intitolato «The Sign of the Orc», il quale, oltre che dare alcuni utili suggerimenti, è indispensabile per giocare in quanto su di esso è basato il sistema di protezione dalla copia del programma.

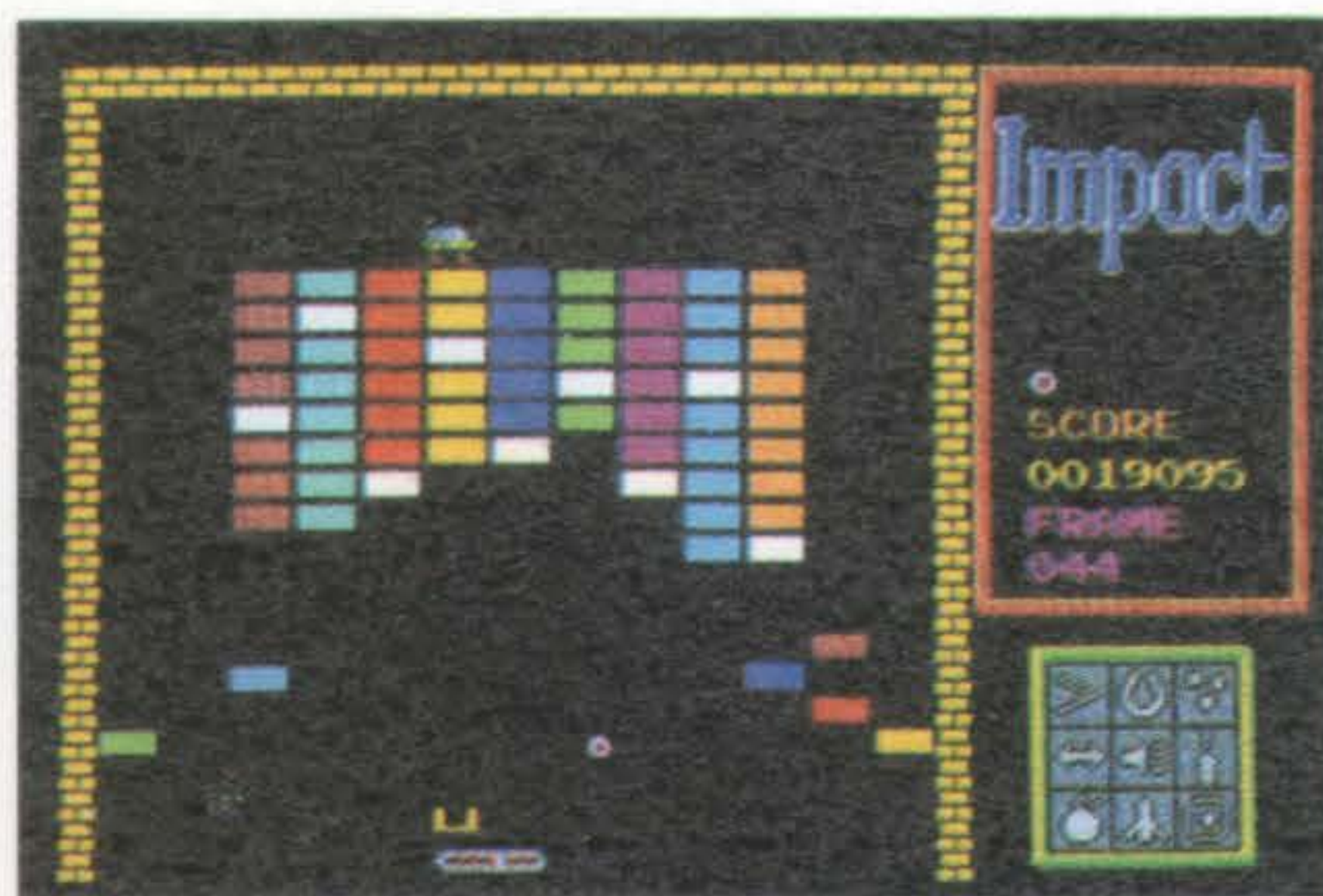
SUGGERIMENTI E TRUCCHI VARI

a cura di GIULIO BONIFAZI

Se sbagliate a scrivere un comando mentre siete in CLI, non è necessario battere il backspace fino alla completa cancellazione: basta usare ctrl-x e l'intera linea sparirà. Se volete eseguire più comandi del Dos, battetene uno, quindi premete ctrl-j; il cursore si sposterà su una nuova linea: battete ancora ctrl-j e scrivete così tutti i comandi che volete far eseguire. Alla fine premete return.

Se volete mantenere in cima allo schermo qualche cosa che vi interessa ed il cursore è vicino al bordo inferiore, battete ctrl-k più volte ed il cursore salirà; battete return per far riapparire il prompt vicino al cursore. Per pulire lo schermo, battete ctrl-l.

Avete la necessità di creare batch files in AmigaDos? Può risultare utile l'uso del Type piuttosto che dell'Echo, che può andare bene per far scrivere una linea, ma ha lo svantaggio di dover essere caricato dal computer ogni volta che viene incontrato. Per usare il type, basta creare il file da far apparire sullo schermo con un qualsiasi editor come ED, e nel batch file scrivere:....type <nomefile>; il testo verrà scritto in maniera molto veloce.

Un utilissimo trucchetto per l'avventura «Leisure suit Larry in the land of the longe lizards», la cui soluzione completa è su questo stesso fascicolo nella rubrica dedicata a questo genere. Come certo saprete, per poter iniziare il gioco vero e proprio è necessario rispondere a quattro domande, ogni volta differenti. Non tutti sono in grado di dare le risposte esatte, ma c'è un modo molto semplice per saltare in blocco i quesiti: quando appare la prima domanda con le quattro alternative possibili A, B, C e D, premete il tasto ALT più il tasto X e magicamente inizierete la vostra avventura; se poi proprio non riuscite a venirne a capo, andate a leggere la soluzione a pagina 14!

Ecco tutte le password per accedere direttamente ai vari livelli di «Impact», gioco ispirato al «Breakout» e parente di «Arkanoid». Ad ogni codice corrisponde l'ingresso automatico ad un determinato livello, perciò battete quello relativo che desiderate, ricordando che USER serve per giocare con gli schermi editati e costruiti da voi:
LIVELLO 11 = GOLD
LIVELLO 21 = FISH
LIVELLO 31 = WALL
LIVELLO 41 = PLUS
LIVELLO 51 = HEAD
LIVELLO 61 = FORK
LIVELLO 71 = ROAD
LIVELLO 81 = USER

Come saprete certamente (oppure no?) nel Kickstart ci sono dei messaggi nascosti che appaiono soltanto se vengono premuti particolari tasti in una esatta sequenza. Seguite passo passo quanto segue: una volta caricato il Workbench, premete contemporaneamente i due shift ed i due alt; tenendoli sempre premuti, schiacciate F1: nella barra dei menu vedrete il primo messaggio. Ad ogni tasto funzione corrisponde una scritta diversa. Quando li avrete visti tutti, tenete attivi ancora i quattro tasti, i due shift ed alt, più F1: togliete il dischetto dal drive SENZA TOGLIERE MAI LE DITA DALLA TASTIERA e magicamente leggerete un altro messaggio; inseritene un altro diverso, ovviamente facendovi aiutare da qualche amico vista la limitatezza del numero delle dita disponibili, e per qualche secondo apparirà quello più bello, perciò occhio alla barra! Caricate ora le Preferences (potete lasciare quei quattro tasti ora) perché anche qui ci sono dei piccoli scherzetti, anche se più difficili da tirar fuori. Sulla destra dello schermo ci sono i due gadgets che riproducono il mouse: dovete innanzitutto clickare sul gadget sinistro, prima sulla riproduzione del pulsante di sinistra e poi su quella di destra, poi passate all'altro gadget e clickate anche qui sui due pulsanti nello stesso ordine. Ripetete questa operazione 4 VOLTE per un totale di 16 click. Ora selezionate il Change Printer, quindi attivate la seriale. Portatevi sopra la freccia che punta verso l'alto, normalmente usata per scegliere il driver della stampante, e clickate finché non verrà fuori un romantico messaggio sulla solita barra.

Tutti i possessori della stampante Okimate 20 saranno ben felici di leggere quanto segue e chissà che qualcun altro non la compri appositamente! La peculiarità di questa stampante è di lavorare con un particolare tipo di nastro nero o a colori a trasferimento termico. Sfruttando questa caratteristica è possibile imprimere

un qualsiasi disegno di Amiga su di una maglietta di cotone bianca (non certo inserendola al posto della carta!). Tutto quello che dovete fare è stampare normalmente la schermata su carta, metterla girata sulla maglietta e passarci sopra il ferro da stiro. Il risultato è davvero notevole, provare per credere!

Magari avete necessità di crearvi dello spazio su un disco Workbench per inserire qualche programma di utilità, per esempio uno shell od un nuovo editor.

In questo caso è indispensabile cancellare dei files dal disco wb (ricordarsi di lavorare SEMPRE su una copia!): i più «sacrificabili» sono sicuramente i programmi Demo (dots, lines, etc.) ed i drivers delle stampanti che non si utilizzano.

Questi drivers si trovano nella directory devs/printers e potete cancellare tranquillamente quelli che non vi servono.

Se per esempio avete una stampante Epson, battete:

```
makedir ram:
cd devs
copy printers/epson to ram:
delete printers all quiet
makedir printers
copy ram: to printers
```

battete ora un «dir printers» per controllare che il driver sia stato copiato correttamente e che sia presente.

Capita di non dover caricare sempre il Workbench ma di volersi fermare soltanto in Cli: affannosamente premiamo Control D per fermare la startup-sequence ed impedire quindi il LOADWB. C'è un metodo sicuramente migliore che consiste nell'inserire una domanda apposita del tipo «Carico il Workbench ? (Y oppure return)» e gestire la risposta. Aggiungete quanto segue alla vostra startup-sequence editandola con l' Ed (ed s/startup-sequence):

```
ECHO "Carico il Workbench ? (S oppure Return)"
FAILAT 25
MAKEDIR > NIL:?
FAILAT 10
IF EXISTS s THEN
DELETE s
LOADWB
ENDCLI > NIL:
ENDIF
```

State molto attenti a non battere nient'altro che una sola S se volete caricare il Workbench, altrimenti la routine non funzionerà e vi troverete con una directory non voluta sul dischetto! Difatti il funzionamento è molto semplice: in pratica quello che battete al prompt viene preso come nome per creare una nuova sottodirectory ed alla linea successiva ne viene controllata l'esistenza o meno: se c'è, viene immediatamente cancellata e caricato il Workbench, altrimenti vi troverete in Cli. ■

ISTRUZIONI E VARIABILI

2ª parte

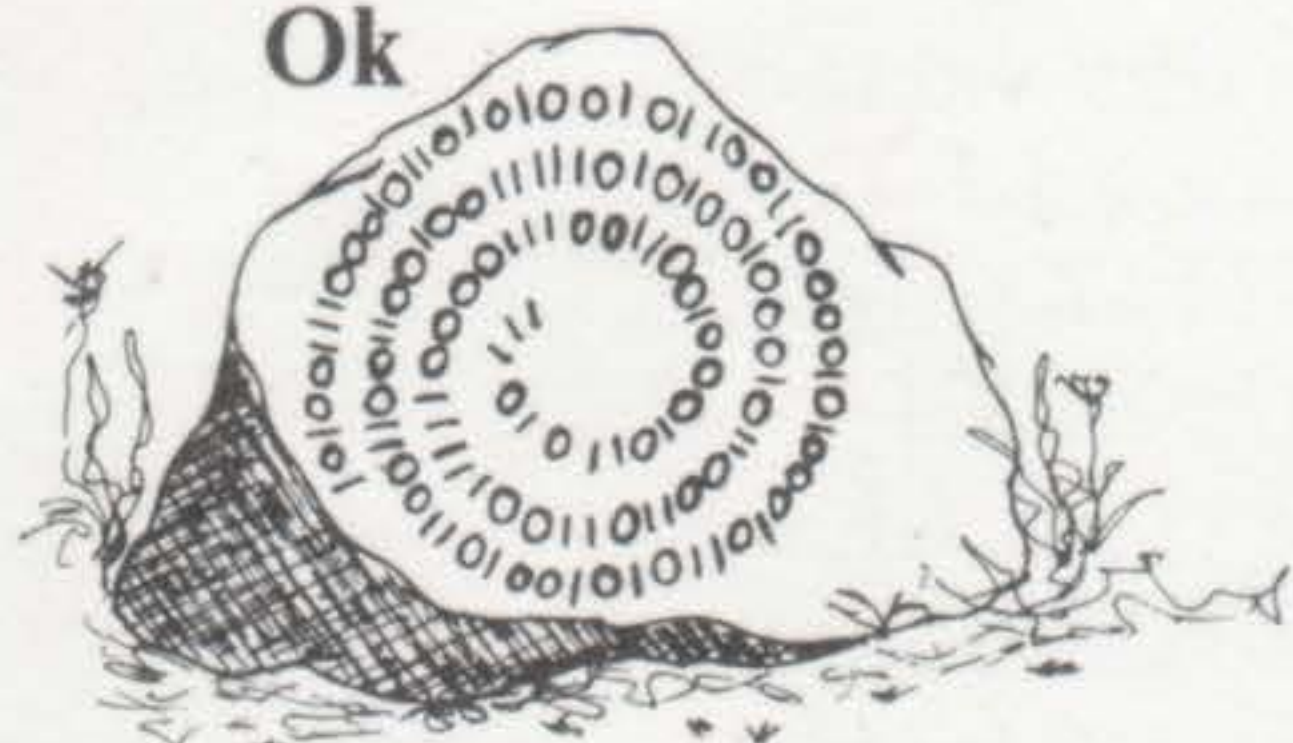
di LUCA BRIGATTI

Si è già detto della differenza che sussiste fra comandi e variabili; i primi vengono eseguiti in modo diretto, ovvero immediatamente dopo essere stati digitati; le altre vengono raccolte in un programma ed eseguite tutte insieme. In realtà tale differenza è più teorica che pratica; la maggior parte delle istruzioni infatti può essere eseguita come un comando e viceversa.

Facciamo un esempio pratico con una delle istruzioni più utili e sfruttate del Basic: l'istruzione PRINT, che consente di scrivere qualcosa sullo schermo. Provate a scrivere nell'output window (quella che accoglie i comandi e visualizza l'output dei programmi):

```
PRINT "5+4 fa " [RETURN]
PRINT 5+4 [RETURN]
```

otterrete il seguente output sullo schermo:

```
PRINT "5+4 fa "
5+4 fa
Ok
PRINT 5+4
9
Ok
```

Come si vede, i due comandi sono stati eseguiti subito, e il calcolatore ci avverte dopo ogni comando che tutto è andato bene con il prompt «Ok».

Se le due istruzioni di cui sopra vengono scritte nella List Window (quella che accetta le istruzioni e contiene il listato di un programma) non succederà niente di particolare ed esse rimarranno lì dove sono finché non verrà dato il «Run» (da menu oppure scrivendo «Run» nell'output window o premendo CTRL-R); allora la List Window sparirà, l'output window verrà pulita e compariranno le scritte:

```
5+4 fa
9
Ok
```

indicative del fatto che le nostre istruzioni sono state eseguite e che l'esecuzione delle stesse non ha dato problemi («Ok» alla fine).

Giacché non vi è, come volevasi dimostrare, quella grande differenza fra istruzioni e comandi, d'ora in poi con il termine istruzioni, a meno che non venga specificato altrimenti, intenderemo riferirci sia alle istruzioni vere e proprie che ai comandi.

In questi esempi sono stati toccati due punti importanti.

Come si è visto, di seguito all'istruzione PRINT è stato posto un *argomento*, ovvero un «qualcosa» che completa l'istruzione e ne specifica e dirige l'azione.

Ciò, in definitiva, è più che intuitivo; se diciamo al computer «Scrivi!» dobbiamo anche indicargli *cosa* scrivere.

La maggior parte delle istruzioni richiede un argomento; per alcune però è opzionale, avendo queste ultime un'azione anche in mancanza di esso.

Secondo punto: nella prima istruzione PRINT l'argomento ("5+4 fa") è racchiuso fra virgolette; nella seconda no, ciò causa un funzionamento molto diverso dell'istruzione: *quando un argomento è racchiuso fra virgolette, l'istruzione PRINT si limita a riportarlo tale e quale, previa eliminazione delle virgolette stesse; quando un argomento non è racchiuso fra virgolette,* il che accade generalmente quando trattasi di espressioni (come appunto 5+4) o di variabili, *l'istruzione «valuta» (cioè risolve) l'espressione e scrive il risultato.*

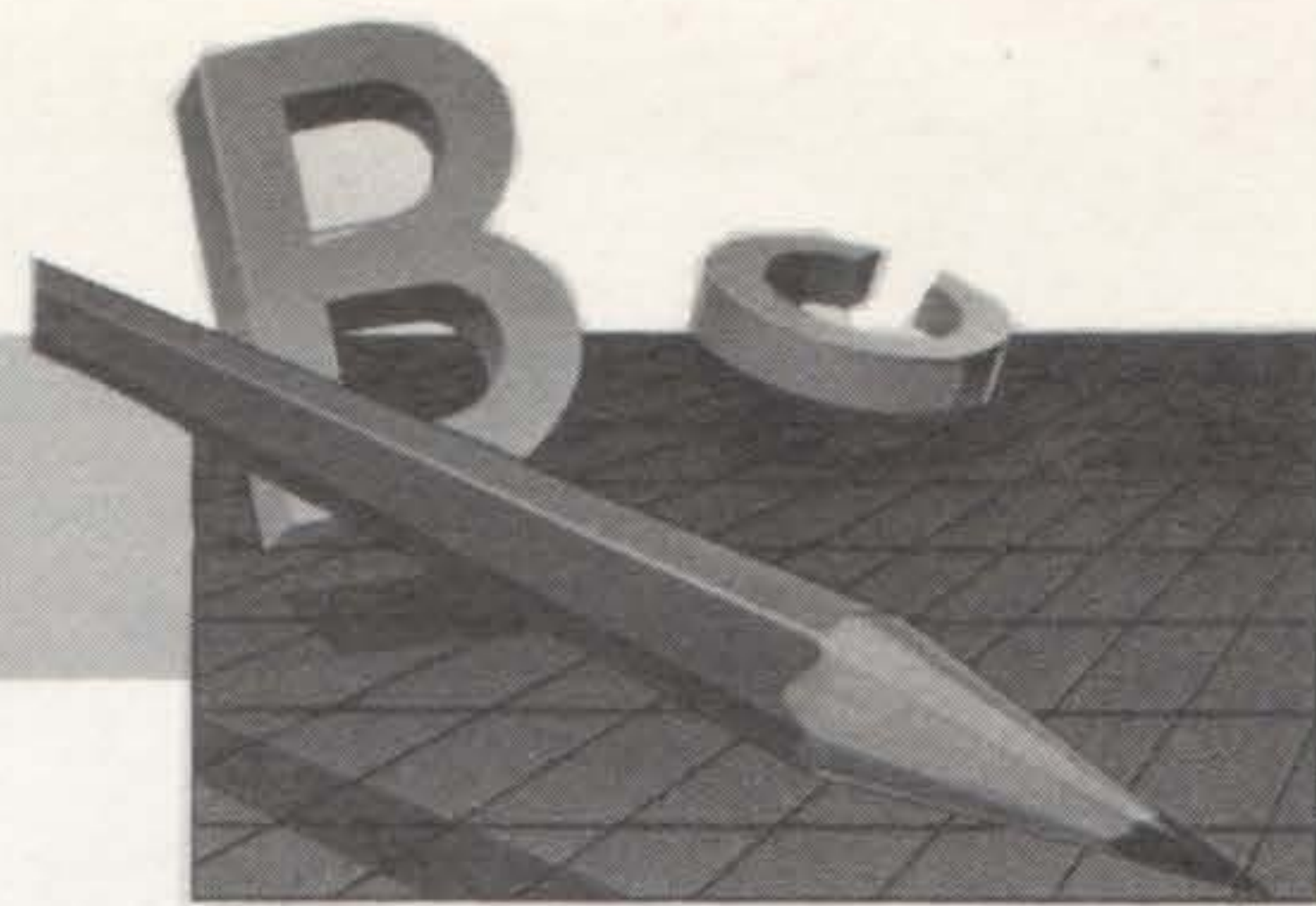
Se vi sentite pigri, potete utilizzare il punto di domanda «?» al posto di scrivere «PRINT»; per il sistema fà lo stesso. Un'ultima nota: al termine di ogni istruzione occorre premere il tasto «RETURN». Da ora in poi ciò verrà dato per scontato ed eviteremo di ripeterlo ogni volta. Ricordatevene.

SERENO VARIABILE

Ed arriviamo al terzo componente dell'eterno triangolo Comandi-Istruzioni-Variabili.

Chi mastica un po' di algebra elementare avrà avuto occasione di imbattersi in equazioni del tipo:

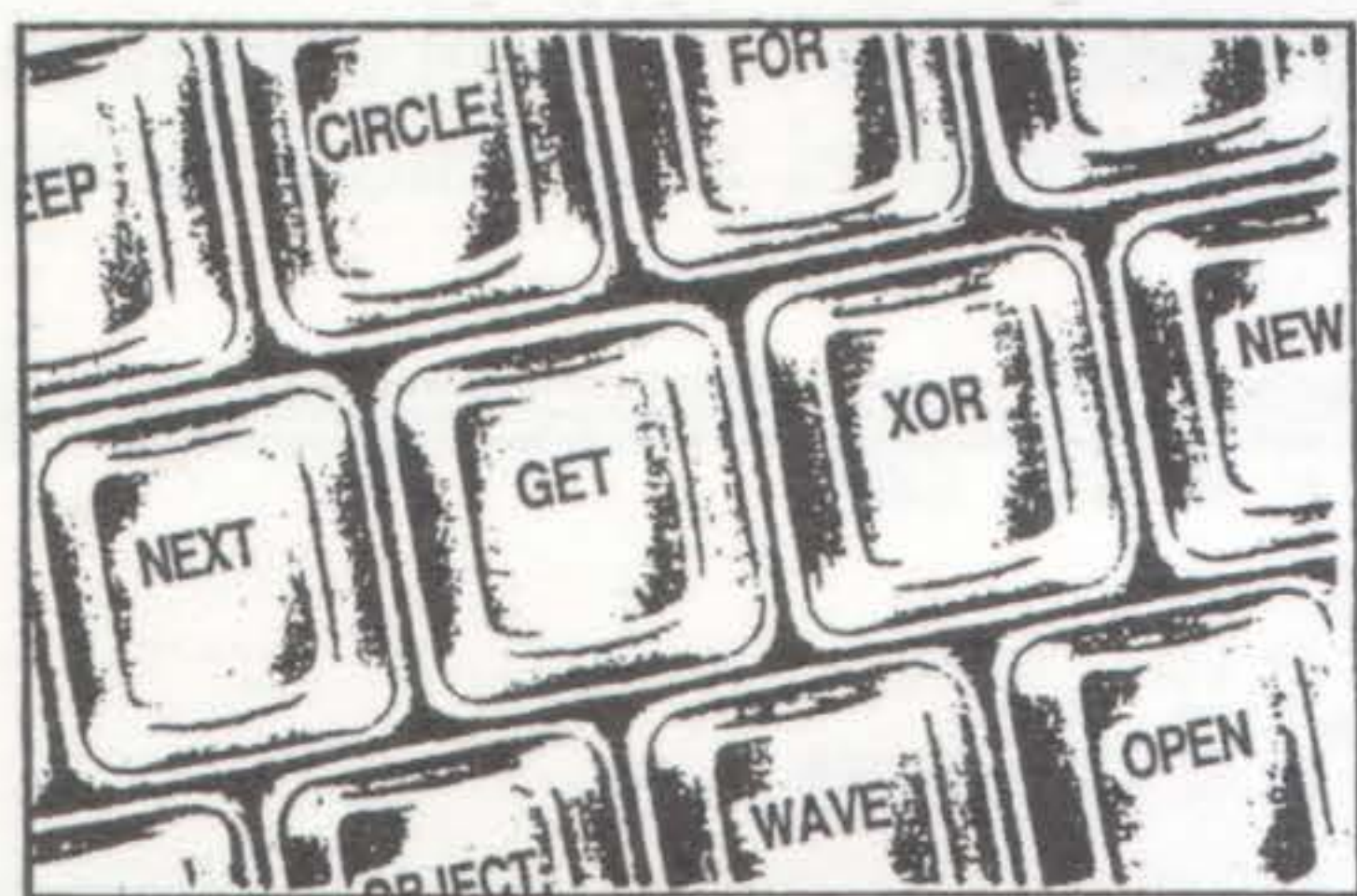
y=2*x+3

(l'asterisco indica, in Basic, il segno di moltiplicazione).

Ebbene, «x» e «y» sono variabili, ossia valori numerici variabili indicati da una lettera.

In Basic le variabili possono avere nomi di più lettere, massimo quaranta ed i nomi possono essere composti anche da numeri e dal carattere punto (.) purché non come iniziale.

Sono nomi validi, ad esempio:
Iva ; Totale ; Sconto ; Importo ; Dato1 ; Dato2 etc...

Il computer non fà differenza fra lettere maiuscole e minuscole, così «TOTALE», «Totale» e «totale» sono a tutti gli effetti la stessa variabile.

Per chi sente menzionare le variabili per la prima volta in questo contesto, diremo che esse possono essere immaginate come delle scatole, ognuna delle quali ha un nome ben preciso, contenenti ciascuna un numero che può essere modificato (da qui il termine variabili) nel corso del programma. Per il programma, *l'uso di una variabile corrisponde a tutti gli effetti all'uso del numero che essa rappresenta.*

Esistono quattro tipi di variabili numeriche, in relazione al tipo di valore che esse possono contenere; questi tipi sono:
Variabili intere in singola precisione: contengono valori *interi* (cioè senza virgola) compresi fra —32768 e 32767.
Variabili intere in doppia precisione: contengono valori interi compresi fra —2.147.483.648 e 2.147.483.647.
Variabili reali in singola precisione: contengono valori *reali* (cioè con la virgola, ma anche valori interi) compresi fra $1{,}18 \times 10^{-38}$ e $3{,}4 \times 10^{38}$.
Variabili reali in doppia precisione: contengono valori reali compresi fra $2{,}23 \times 10^{-308}$ e $1{,}79 \times 10^{308}$.
Per informare il computer sul tipo di variabile che si intende definire, è necessario posporre al nome della variabile un carattere particolare, diverso per ogni tipo di variabile. Tali caratteri sono:

Intero in singola precisione = %
Intero in doppia precisione = &
Reale in singola precisione = !
Reale in doppia precisione = #

Ad esempio, potremo avere variabili con nomi tipo:
Dato% ; Totale& ; Distanza! ; Tempo# , etc.

Se al nome della variabile non viene fatto seguire alcunché, il sistema la considererà di tipo reale in singola precisione, ovvero le variabili «Distanza» e «Distanza!» sono a tutti gli effetti equivalenti.

Nella maggior parte dei casi le variabili reali in singola precisione si adattano perfettamente agli scopi del programmatore, pertanto non vale la pena di angustiarsi nella scelta del tipo di variabile da usare. In casi particolari, ad esempio quando serva una precisione di calcolo maggiore, si userà la doppia precisione; quando serva una velocità di calcolo maggiore sarà opportuno usare variabili intere, possibilmente in singola precisione; anche quando occorra risparmiare memoria, ad esempio trattando una grande massa di dati, le variabili intere e la singola precisione sono più indicate.

ALLACCIAMOCI LE STRINGHE

Le variabili però non sono soltanto numeriche; esistono infatti anche le variabili alfanumeriche, dette *stringhe* (chissà poi perché) in linguaggio informatico.

Esse sono caratterizzate dal simbolo dollaro («$») fatto seguire al nome, che per il resto rispetta le convenzioni viste per le variabili numeriche.

Esempi di variabili stringa sono:
Nome$, Titolo$, Data$, Tipo$, etc...

Tali variabili possono contenere tutti i caratteri (anche quelli non visualizzabili) che il computer è in grado di gestire. Se sulle variabili numeriche possono essere eseguite tutte le operazioni consentite per i numeri, sulle stringhe ciò non è possibile; due stringhe possono solo essere sommate ed il risultato è una stringa uguale alla prima seguita dalla seconda. Ad esempio, se si sommano le variabili «Nome$» contenente la stringa «Mario» e «Cognome$» contenente la stringa «Rossi», il risultato sarà la stringa «Mario Rossi».

QUESTIONE DI DEFINIZIONI

Per *definire* una variabile, ovvero assegnarle un valore, si usa l'istruzione LET.

Ad esempio le istruzioni:
LET Totale=200:LET Iva=18:LET Raggio%=4000
LET PiGreco=3.14:LET Area#=
PiGreco*Raggio%*Raggio%
creano le variabili (reali in singola precisione) «Totale», «Iva», «Raggio%», «PiGreco» e «Area#» ed assegnano loro i valori numerici (rappresentati da numeri, variabili o espressioni) che si trovano dopo il segno di uguale;
LET NOME$="Mario":LET Cognome$="Rossi"

assegna i valori «Mario» e «Rossi» alle variabili Nome$ e Cognome$ rispettivamente.

Si noti l'uso delle virgolette per le variabili stringa, che

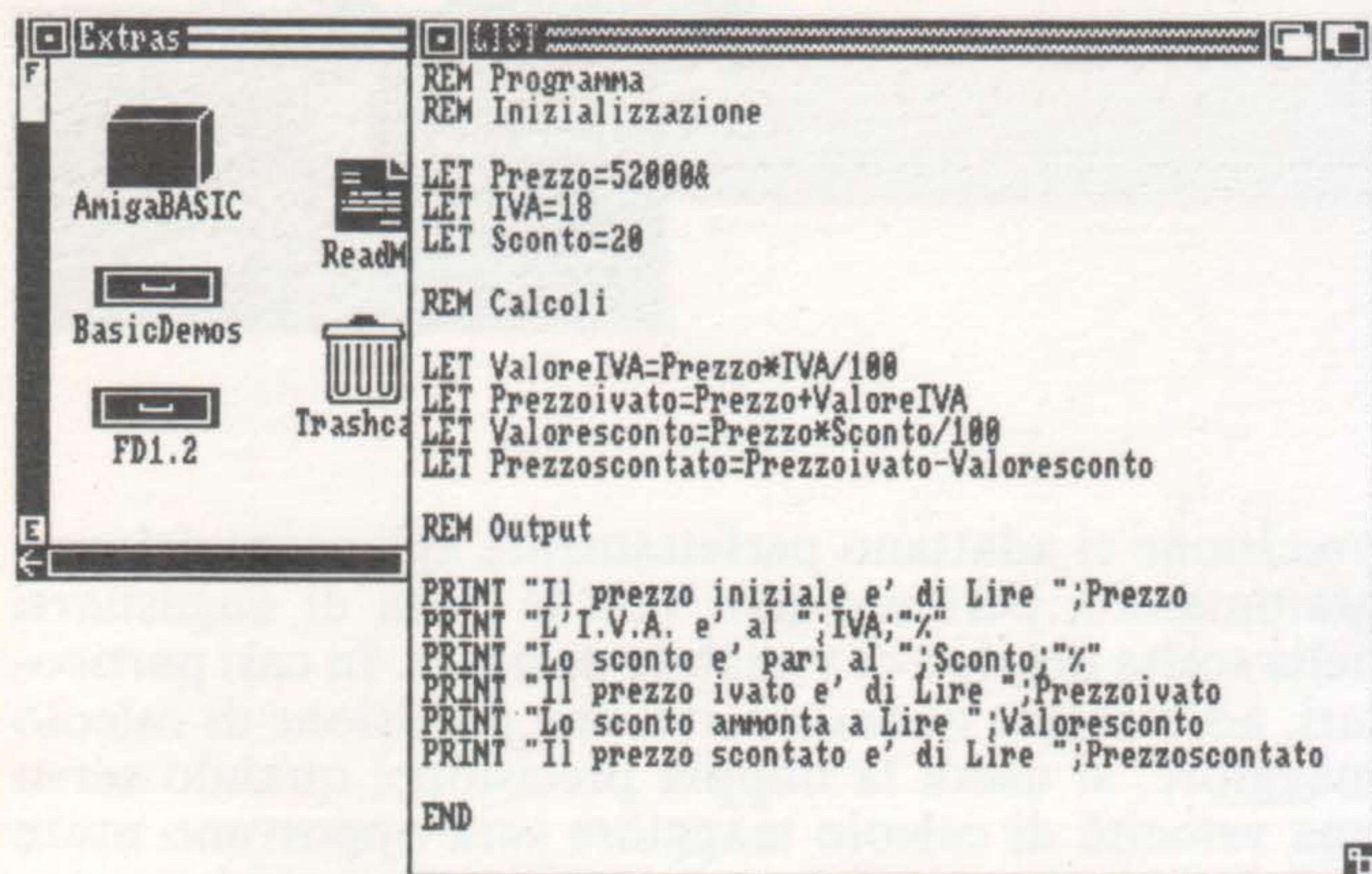



Fig. 1 - Programma che calcola un prezzo maggiorato di Iva e scontato.

invece mancano nella definizione delle variabili numeriche.

Si noti inoltre l'uso dei due punti «:» per scrivere sulla stessa linea di programma più istruzioni separate.

L'istruzione LET, unica eccezione in Basic, può essere omessa, così le due seguenti:

LET Totale=200
Totale=200

sono equivalenti ed entrambe ammesse.

Se un'istruzione utilizza una variabile non precedentemente definita, il sistema assume che essa abbia valore 0 (si dice che il valore di *default* è 0) se è una variabile numerica e *stringa nulla* (ovvero vuota) se è una variabile stringa.

In Basic, quando si ha un numero con la virgola, occorre utilizzare il *punto decimale*, in modo conforme alla notazione aritmetica anglosassone, e non la virgola, come nel sistema italiano; così

LET PiGreco=3,14

è scorretto e verrà generato un messaggio d'errore mentre

LET PiGreco=3.14

è corretto e verrà accettato.

UN PROGRAMMA ESEMPLIFICATIVO

Scriviamo ora un programma che faccia uso delle istruzioni esaminate.

Si voglia un programma che, dati il prezzo di un capo, il valore dell'Iva e l'ammontare percentuale dello sconto applicato al capo, calcoli il costo finale dello stesso, maggiorato dell'Iva e scontato.

Il programma che fà al caso nostro si trova in Figura 1.

Analizziamolo insieme. Esso inizia con un'istruzione nuova, REM, abbreviazione di REMember o REMind (letteralmente «Ricorda» o «Tieni a mente»); l'argomento di tale istruzione è una stringa di caratteri alfanumerici che costituisce un messaggio a chi legge il programma.

Il sistema ignora completamente tale istruzione, come se non esistesse, il che induce molti programmatori a non farne affatto uso.

Tale istruzione è invece utilissima. Un programma che contenga molte REM in punti strategici è molto più leggibile (e comprensibile) di un programma che non ne contenga affatto. Un consiglio dunque: usate abbondantemente tale istruzione, per agevolare sia chi leggerà il vostro programma che voi stessi, se vi dovesse capitare di riprenderlo in mano dopo mesi od anni.

Il primo segmento del programma si occupa di *inizializzare* le variabili, ovvero di assegnare loro un valore per la prima volta. La seconda parte del programma definisce altre variabili in base ai risultati di calcoli sulle variabili precedenti.

Notate che in questi calcoli compaiono insieme variabili e numeri, anche detti *costanti*.

Si noti inoltre che in Basic il simbolo per la divisione è la barra di frazione «/».

Segue poi la fase di visualizzazione dei risultati.

Come si nota, l'argomento di una stessa istruzione Print può essere composto da stringhe o numeri (in questo caso da variabili), separati dal punto e virgola «;» che fa sì che l'istruzione Print scriva gli argomenti di seguito, sulla stessa linea. Non ci sorprenda inoltre constatare

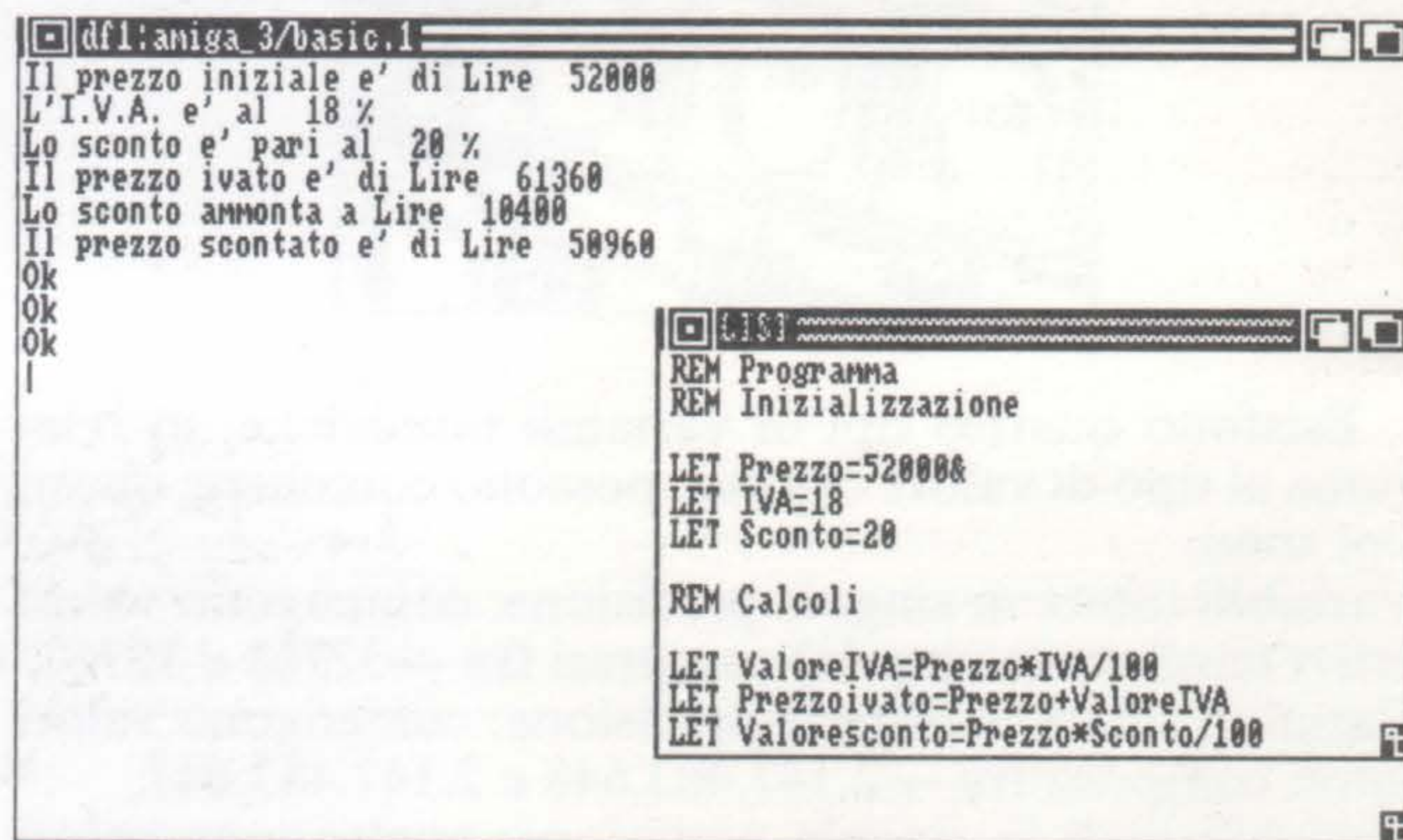



Fig. 2 - Output del Programma di Figura 1.

che quando nell'argomento compare il nome di una variabile (beninteso non racchiusa fra virgolette), viene stampato il suo valore numerico. Il programma termina con l'istruzione END (Fine). Tale istruzione informa il sistema che il programma è terminato. Il sistema quindi si comporta di conseguenza terminando eventuali operazioni ancora attive e restituisce il controllo al programmatore.

A rigore questa istruzione non è indispensabile, ma è bene metterla comunque.

Provate a modificare il programma cambiando il valore delle variabili, aggiungendo altre istruzioni Print o modificando il tipo di variabili usate, posponendo ai loro nomi i simboli visti più sopra, ed osservate i risultati. ■

Per DIR di più...

Dalla directory principale si dipartono, per così dire, altre directory. Che possono contenere files o altre directory ancora. Esaminiamo il contenuto di queste ultime.

di RICKY SWORD

Abbiamo visto in precedenza come esaminare il contenuto di un disco tramite il comando Dir.

Esso, di per sé, si limita a mostrarci il contenuto della *directory principale* (in inglese *Root directory*) elencando il nome dei files e delle altre directory presenti sul disco. Ma, come si è detto, anche le directory contengono files e possono contenere altre directory; come esaminare dunque anche il contenuto di queste ultime? Da Workbench è molto semplice, se esse hanno un'icona che le rappresenti, generalmente raffigurante il cassetto: basta attivare l'icona stessa (portando il puntatore del mouse sull'icona e premendo due volte il tasto sinistro dello stesso) e *voilà*, ecco comparire una finestra nel cui interno appaiono le icone dei files e delle subdirectory appartenenti alla directory in esame.

Se invece la directory da esaminare non possiede l'icona, essa non risulta visibile da Workbench, né tantomeno lo è il suo contenuto. Da CLI invece è sempre possibile esaminare il contenuto di una directory semplicemente specificando il suo nome dopo il comando Dir; ad esempio:

Dir c
mostra il contenuto della directory «c» che contiene i comandi del Dos; invece:

Dir devs
mostra il contenuto della directory «devs» che contiene files necessari al sistema per pilotare le periferiche (ad esempio stampanti, modem, tastiera).

Naturalmente è possibile esaminare anche la subdirectory; per farlo basta scrivere, dopo il comando Dir, il nome della directory e della subdirectory separate dalla barra di frazione «/»; ad esempio:

Dir devs/printers
mostra i files che pilotano le stampanti, contenuti in «printers», contenuta a sua volta in «devs».

Per esaminare il contenuto di un altro disco, invece, occorre specificare il nome del disco stesso seguito dai due punti. Così, per il disco «Extras», si scriverà:

Dir Extras:
A questo punto appare un requester chiedendo di inserire (in un drive qualsiasi, nel caso se ne abbia più di uno) il disco «Extras»; fatto ciò, compare il contenuto del disco in questione.

Se possedete un solo drive e date di nuovo il comando:

Dir Extras:
vi verrà chiesto di inserire il Workbench quindi, dopo un secondo o due in cui si accende la rossa spia del drive, di nuovo il disco Extras e solo allora ne vedrete apparire il contenuto. La ragione di ciò è molto semplice: poiché tutti i comandi del Dos si trovano nel Workbench (e precisamente nella directory «c») il sistema, prima di poter eseguire un qualsivoglia comando, lo deve caricare in memoria e, per poterlo fare, deve ovviamente avere a disposizione il Workbench. Tale operazione di caricamento deve essere ripetuta per tutti i comandi, anche se sono già stati impartiti.

Avendo due drives è opportuno tenere sempre il Workbench in uno di essi. Naturalmente si potranno esaminare anche eventuali directory presenti sul disco. Ad esempio:

Dir Extras:BasicDemos
mostrerà il contenuto di BasicDemos.

Riassumendo, per far sì che il comando Dir agisca su di una particolare directory, occorre sempre specificare il *percorso* necessario a raggiungerla, e ciò vale anche per i files e per tutti i comandi del Dos.

Ci si può riferire ad altri dischetti anche specificando il drive nel quale questi si trovano. I nomi dei drive, che possono essere massimo quattro, sono: df0: (quello interno), df1:, df2: e df3: gli altri, per cui:

dir df0:
mostrerà il contenuto del disco che si trova nel drive 0, e

dir df1:
il contenuto del disco nel drive 1, ammesso che sia presente.

Al comando Dir possono essere fatte seguire alcune *opzioni* che ne modificano il comportamento; per farlo occorre aggiungere al comando la dicitura «opt x» con «x» a rappresentare l'opzione desiderata; ad esempio:

dir opt a
esegue il comando dir con l'opzione «a».

Le opzioni disponibili sono:

A (All) Mostra il contenuto della directory corrente e di tutte le directory e subdirectory in essa contenute.

D (Directory) Mostra solo i nomi delle directory, trascurando i files.
I (Interattivo) Entra in modo interattivo.

Nel modo interattivo viene mostrato un oggetto alla volta (directory o file che sia), seguito da un punto interrogativo con il quale il sistema ci chiede come procedere.

Ci sono diversi modi di rispondere: premendo solo RETURN si passa ad esaminare l'oggetto successivo; con E (Enter) si entra in una directory; con B (Back) si esce dalla directory tornando in quella di livello superiore; con T (Type) viene scritto sul video un file di testo; DEL (DELete) cancella un file o una directory se questa è vuota ed infine Q (Quit) interrompe l'esecuzione dell'istruzione Dir restituendo il controllo all'utente.

Il punto interrogativo «?» mostra in qualsiasi momento le possibili risposte da fornire. Ovviamente, le risposte che si danno nell'opzione «I» devono essere coerenti con il file al quale si riferiscono; ad esempio «T» va riservata ai files di testo ed «E» alle directory, pena la comparsa di messaggi d'errore o di risultati bizzarri.

Al comando Dir possono essere associati percorsi e opzioni liberamente, ad esempio:

Dir devs opt a oppure
Dir Extras: opt d o anche
Dir Extras:BasicDemos opt i

sono tutti corretti; fate solo attenzione agli spazi.

Sperimentate liberamente il comando Dir con tutti i dischetti di cui disponete, ponete soltanto cautela nell'usare il «DEL» dell'opzione «I», in quanto un file cancellato in questo modo viene irrimediabilmente perduto.

METTIAMOCI IN *LIST*

Molto simile, come funzione, a Dir, è il comando List (letteralmente «Lista») che, come il precedente, consente di vedere gli oggetti contenuti in un disco o in una particolare directory. Rispetto al precedente, però, List è più completo, fornendo un maggior numero di informazioni.

Provate a dare il comando.

Come potete notare, gli oggetti presenti sul disco vengono elencati in una sola colonna, non in ordine alfabetico, e con directory e files fra di loro mischiate. Accanto ai files però si possono osservare altre quattro colonne contenenti una serie di informazioni relative agli stessi. Nella prima colonna abbiamo un numero che corrisponde alla lunghezza in byte del file in questione. Quando si tratta di una directory (che, essendo il nome di un settore logico del disco e non un file, per definizione non ha lunghezza) viene indicata con il termine «Dir». Inoltre, se è

```
Workbench Screen
List Keys

2> list KEYS
.info                [  881]      45 rwed 26-Sep-80 12:35:26
c                    [  889]     Dir rwed 03-Nov-87 11:03:07
comandi              [  907]     130 rwed 26-Sep-80 11:41:18
System               [  890]     Dir rwed 20-Sep-87 02:05:38
l                    [  891]     Dir rwed 28-Oct-87 12:04:53
devs                 [  892]     Dir rwed 01-Nov-87 17:50:18
s                    [  893]     Dir rwed 25-Oct-87 01:58:29
t                    [  894]     Dir rwed 26-Sep-80 11:43:49
Preferences.info     [  895]     418 rwed 27-Oct-87 16:10:02
Preferences          [  897]   58152 rwed 14-Apr-87 10:37:46
fonts                [  899]     Dir rwed 14-Apr-87 10:38:57
libs                 [  903]     Dir rwed 14-Apr-87 10:39:21
Utilities.info       [  904]     346 rwed 27-Oct-87 16:09:58
Disk.info            [  887]     394 rwed 27-Oct-87 16:40:51
System.info          [  908]     346 rwed 27-Oct-87 16:10:00
Utilities            [  912]     Dir rwed 09-Oct-87 10:03:14
7 files - 9 directories - 142 blocks used
2>
```

TURBOprint

A500 WB 1.2

Esempio d'uso dell'opzione Keys che mostra anche il numero del blocco del disco nel quale si trova l'inizio del file.

stato creato un file che non contiene dati, al posto della lunghezza comparirà la scritta «empty» (Vuoto). Segue poi una sequenza di lettere che, nella maggioranza dei casi, corrisponde a «rwed». Queste lettere ci danno un'informazione sullo stato dei files stessi; «rwed» sta per «Read Write Execute Delete», cioè «Lettura Scrittura Esecuzione Cancellazione» ed indica ciò che possiamo fare con il file.

Se alcune o tutte queste possibilità fossero precluse a quel file, le corrispondenti lettere sarebbero sostituite da «-».

Ad esempio, un file marcato «rwe-» può essere aggiornato (e), letto (r), eseguito (e) ma non cancellato (manca «d»). Ogni file o directory è per default (cioè a meno che non si specifichi altrimenti) «rwed» anche se il parametro «e» ha un significato soltanto per i files eseguibili (cioè per i comandi o i programmi già compilati in linguaggio macchina) e non, ad esempio, per una directory o un testo o un'immagine.

Nella quarta colonna abbiamo la data e nella quinta l'ora in cui quel file è stato creato. Ovviamente la data di creazione e l'ora hanno un senso solo se si possiede una batteria che mantiene attivo l'orologio ed il calendario aggiornato, o se si ha cura di inserire data ed ora esatta all'inizio di ogni sessione lavorativa.

Esiste poi una sesta colonna, nella maggior parte dei casi vuota, che contiene eventuali annotazioni riguardanti il file. Vedremo in futuro come fare per aggiungere annotazioni al nome dei files o modificarne le caratteristiche, che compaiono nella seconda colonna.

Già da quanto si è detto appare chiaro che se il comando Dir è l'ideale per avere una rapida visione d'insieme di files e directory (per la sua caratteristica di elencare i due separatamente e di ordinare alfabeticamente i files presentandoli su due colonne), il comando List è l'ideale per esaminare i singoli oggetti nella directory principale o in qualsiasi altra. Anche con List è necessario specificare il percorso per arrivare ad una determinata directory, così:

List c

Permette di esaminare la directory «c»;

```
Workbench Screen
List P #?

2> list P #?.info
.info                    45 rwed 26-Sep-80 12:35:26
Preferences.info        418 rwed 27-Oct-87 16:10:02
Utilities.info          346 rwed 27-Oct-87 16:09:58
Disk.info               394 rwed 27-Oct-87 16:40:51
System.info             346 rwed 27-Oct-87 16:10:00
5 files - 10 blocks used
2>
```


Esempio del comando «List P #?.info» del disco Workbench 1.2: in questo modo vengono visualizzati soltanto i files che contengono nel loro nome l'estensione «.info».

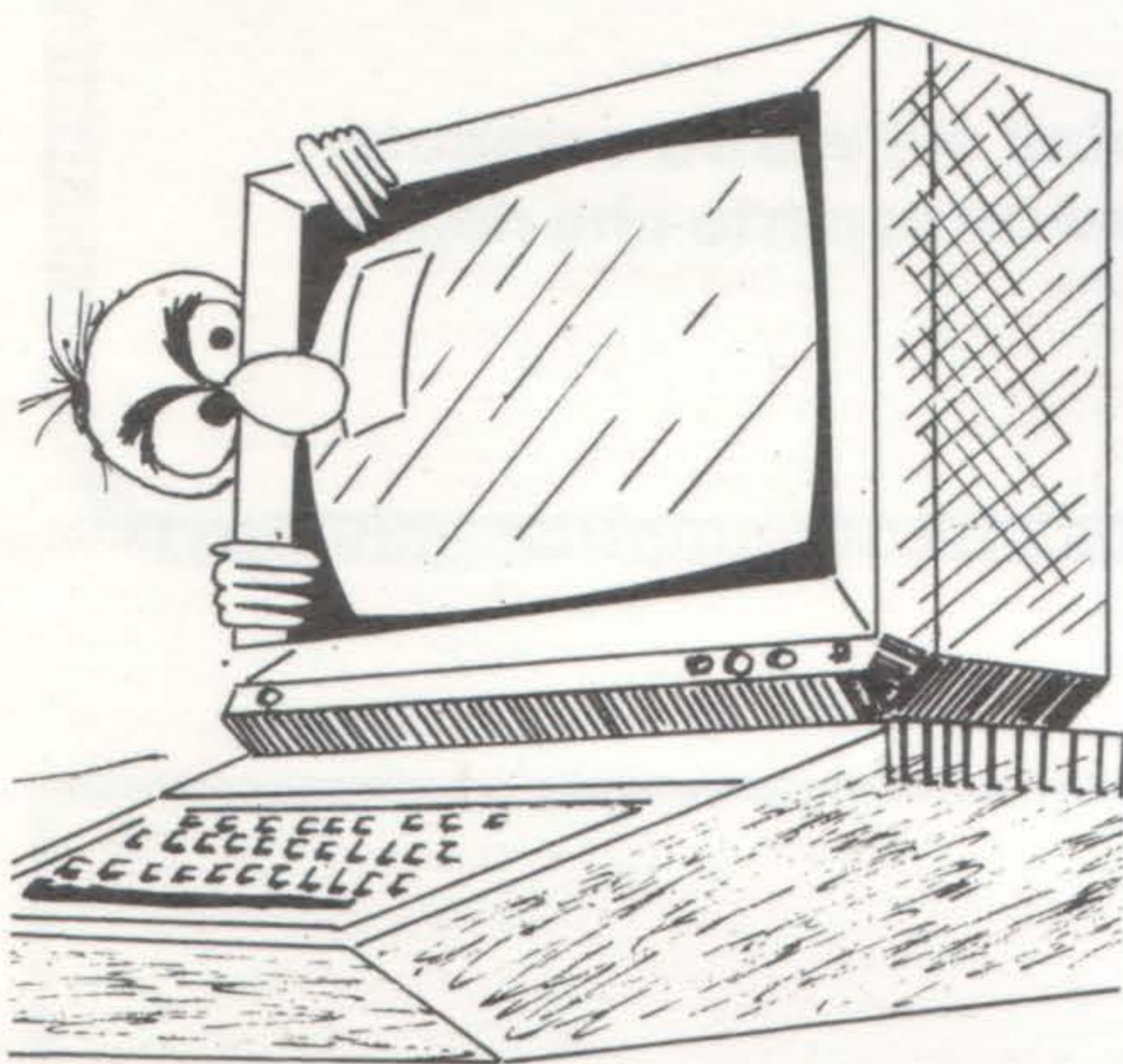
List devs/printers
la directory «printers» nella directory «devs» e

List df1:Basicdemos
la directory «BasicDemos» che si trova nel disco sito nel drive 1.

Anche a List possiamo aggiungere una serie di opzioni, senza far precedere la dicitura «opt».

La prima opzione è DIR <nome> (da non confondere con l'omonimo comando). Se «nome» è il nome di un file, allora verranno visualizzate soltanto le informazioni per quel file; se è invece una directory verrà visualizzato il contenuto di essa (proprio come se l'opzione DIR non ci fosse).

L'opzione TO indica dove deve essere visualizzato l'ouput del comando List:

List to Elenco
Crea un file chiamato elenco che contiene la lista della directory corrente,

List to prt:
Scrive la lista su stampante anziché su video.

L'opzione KEYS mostra anche il numero del blocco del disco in cui si trova l'inizio del file. L'opzione DATES mostra anche la data (e l'ora) di creazioni dei files, proprio come la si avrebbe anche non mettendola (chi può ambire a conoscere la mente dei progettisti di *Amiga*?).
NODATES invece evita che ciò avvenga.
SINCE <data> (letteralmente «DA <data>») è un'opzione molto potente, permettendo di visualizzare solo i files creati a partire dalla data specificata.
UPTO <data> (letteralmente «FINO A <data>»), similmente, visualizza i files creati fino alla data specificata.
S <stringa> («S» è appunto l'abbreviazione di «Stringa») permette di listare solo i files che contengono nel loro nome la stringa specificata.
QUICK (letteralmente «Veloce») mostra una lista limitata ai soli nomi di files e directory, trascurando gli altri dati, come fà il comando «Dir».

Abbiamo lasciata per ultima l'opzione P <tipo> (abbreviazione di Pattern, ovvero «modello») che permette di listare solamente i files che si conformano al modello specificato. Per specificare il modello voluto si utilizzano alcuni caratteri speciali detti «*wildcards*» (letteralmente «Carte Selvagge»; in genere si preferisce mantenere il nome inglese).

Essi sono:

?	Indica ogni singolo carattere.
%	Indica la stringa nulla.
# <c>	Indica un numero qualsiasi (anche zero) di caratteri «c»
<m1> <m2>	Indica tutti i files il cui nome corrisponde al modello 1 seguito dal modello 2.
<m1> ¦ <m2>	Indica tutti i files il cui nome corrisponde al modello 1 o al modello 2.
()	Raggruppa più modelli

Naturalmente tutti i wildcards possono comparire associati gli uni agli altri.

Ecco alcuni esempi:

LIST P Prova?	Lista i files con nomi come ProvaA, ProvaB, Prova1 etc.
LIST P Testo#a	Lista i files con nomi come Testo, Testoa, Testoaa, Testoaaa etc.
LIST P # (AB)	Lista i files con nomi come AB, ABAB, ABABAB, ABABABAB etc.
LIST P Figura (a¦b)	Lista i files con i nomi Figuraa e Figurab
LIST P #?	Lista tutti i files (cioè con nome composto da un numero qualsiasi di caratteri qualsiasi)

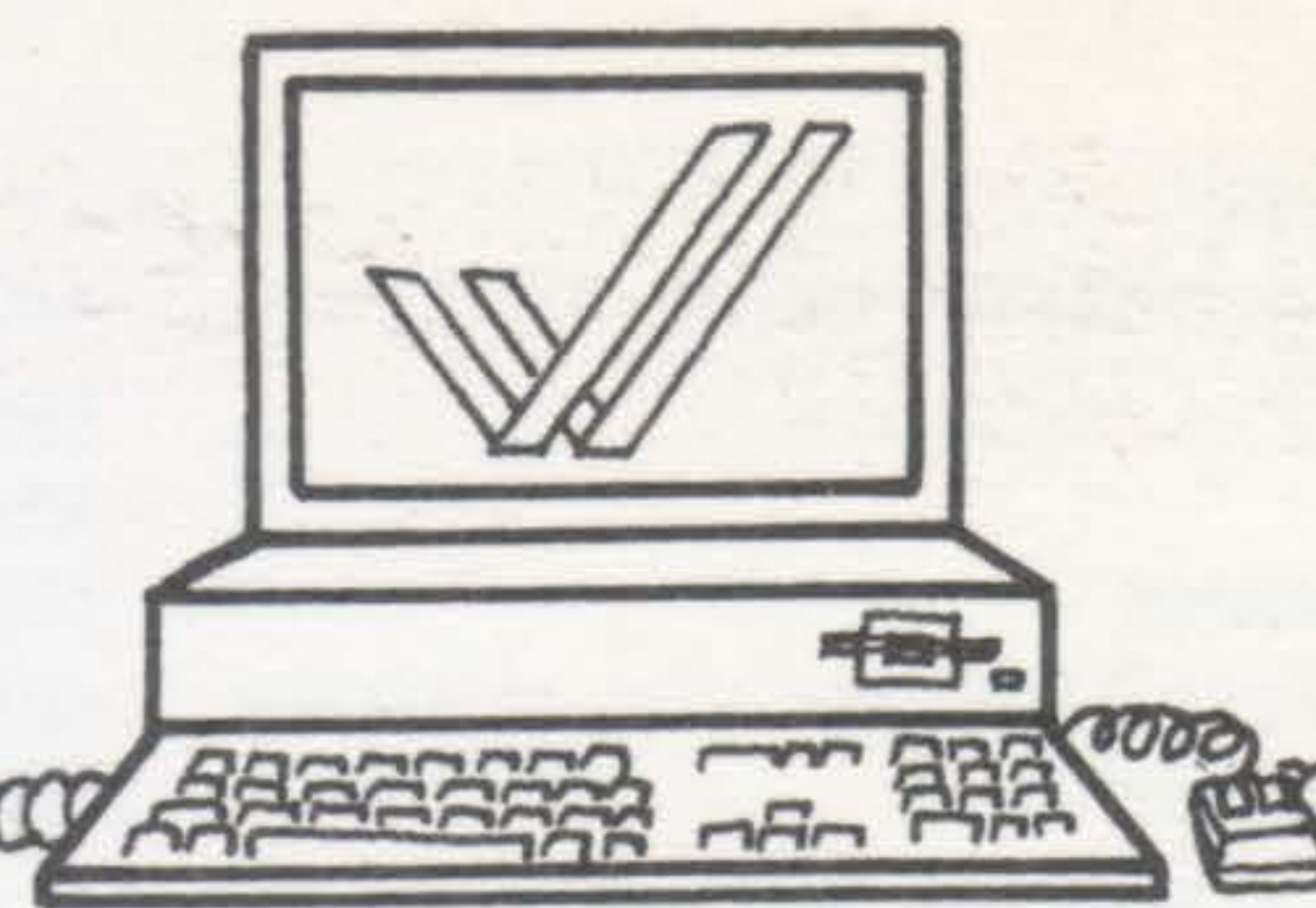
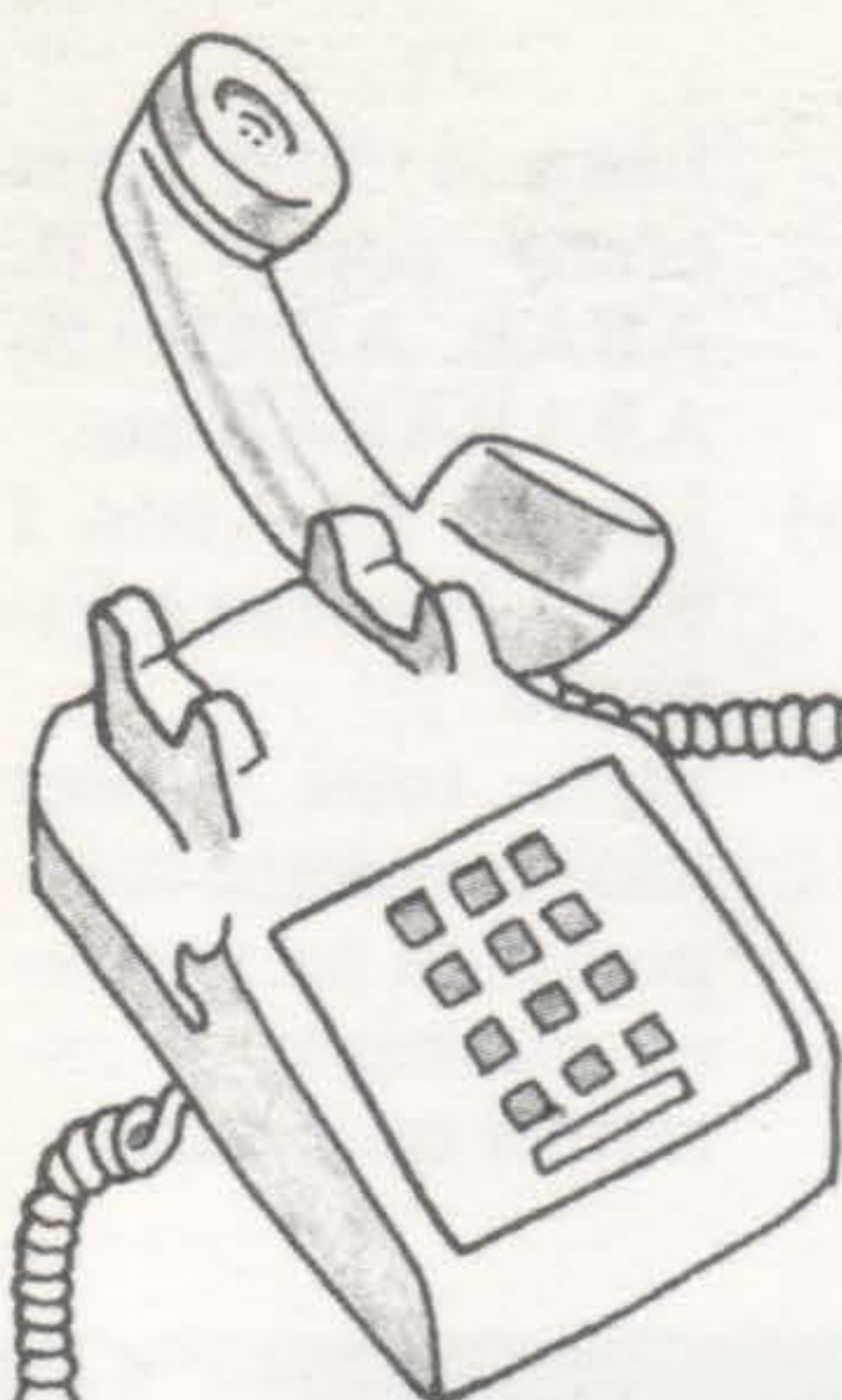
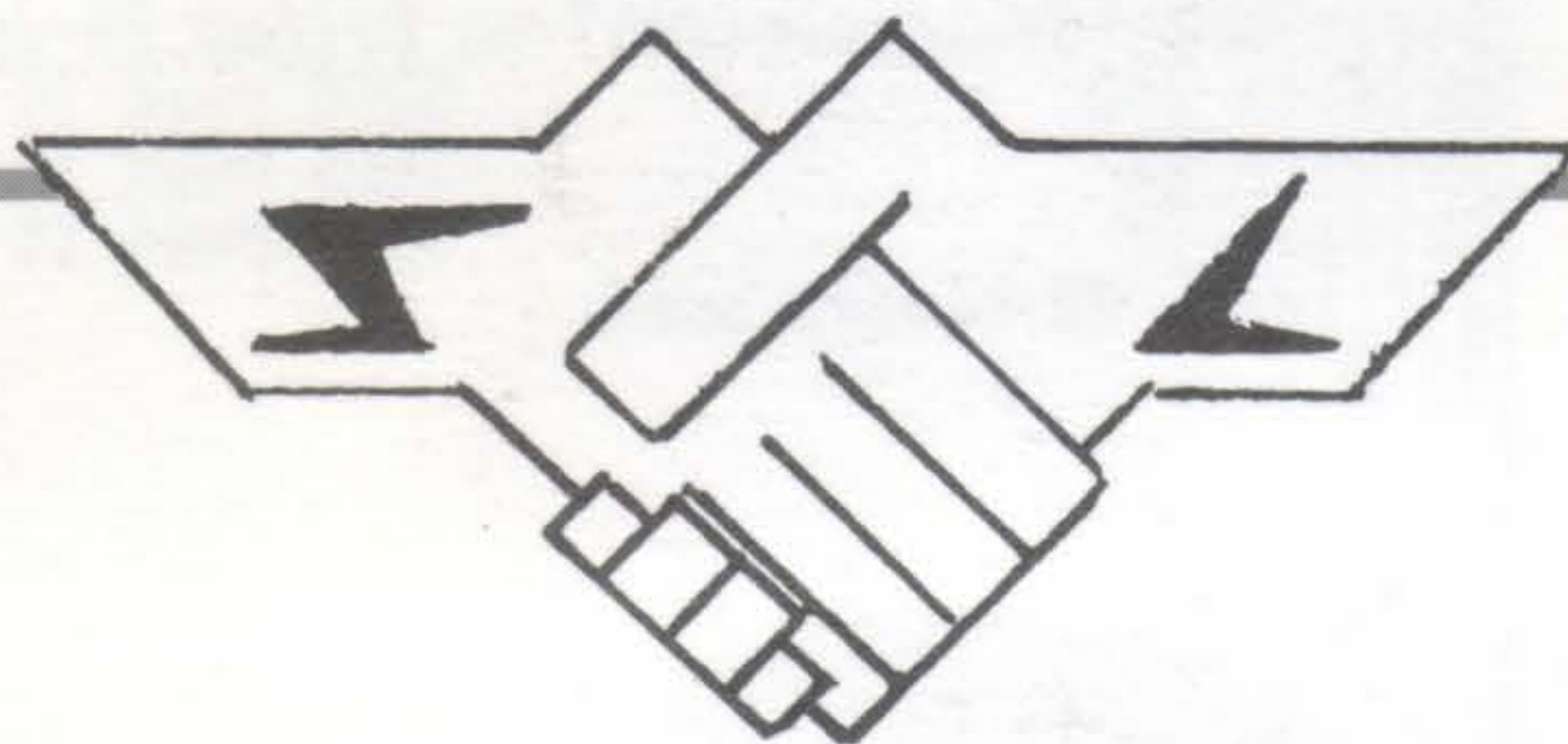
Lista dei files che iniziano con la lettera E ottenuta dando il comando «List P E#?»; per inviare i dati alla stampante piuttosto che al video è stato aggiunto «to prt:».

Elenco delle stampanti presenti nella directory «printers», contenuta nella «devs», che possiedono il suffisso «oki».

La combinata #? corrisponde dunque all'asterisco «*» per l'MS-DOS, quindi:

LIST P A#?	Lista i files il cui nome inizia per A.

Va segnalato che i wildcards funzionano anche per altri comandi, compreso Dir.

Al termine dell'elenco di files ottenuto con List, compariranno l'indicazione del numero di files e di directory contenuti nella directory corrente ed infine il numero di blocchi usati. Tenete presente che un blocco corrisponde a 512 bytes, ovvero a mezzo Kilobyte. In questo caso, come in altri, possono essere indicati contemporaneamente percorsi ed opzioni. Anche con questo comando il nostro consiglio è di sperimentare liberamente con le varie opzioni del comando list, in particolare con i wildcards.

■

PUOI COLLABORARE ANCHE TU

AMIGA Byte è aperta alla collaborazione di tutti quanti fra voi desiderano essere protagonisti oltre che lettori della rivista. Basta conoscere il computer, naturalmente, ed avere idee interessanti o utili per articoli e programmi. Chissà quanti di voi hanno nel cassetto della mente o letteralmente in quello della scrivania programmi realizzati per ottimizzare il proprio lavoro, per occupare intelligentemente il tempo libero, e materiale in genere scaturito dall'esperienza, dall'amore per il proprio fare, dall'inestinguibile sete di sapere e produrre meglio e di più. Be', non teneteli chiusi nel cassetto o nella testa, inviateceli in visione. Tutto il materiale pubblicato sarà regolarmente compensato, il che non guasta, giusto? Spedite sempre una copia dei vostri lavori, dattiloscritti o su disco (l'altra tenetela stretta per sicurezza) specificando sempre i vostri dati. L'ordine e la precisione sono indispensabili. A tutti verrà data risposta, qualunque sia l'esito.
Indirizzate il materiale a Arcadia srl, C.so Vitt. Emanuele 15, 20122 Milano.

SCRIVI DIRETTAMENTE IN REDAZIONE TROVERAI TANTI AMIGHI

A midi

Una semplice interfaccia da autocostruire per controllare strumenti musicali elettronici midi-compatibili con il nostro computer.

di ROLANDO POLIZIANI

Midi come Musical Instrument Digital Interface: è un protocollo (non molto standard, a dir la verità) che rende possibile la comunicazione fra computer ed apparecchi musicali.

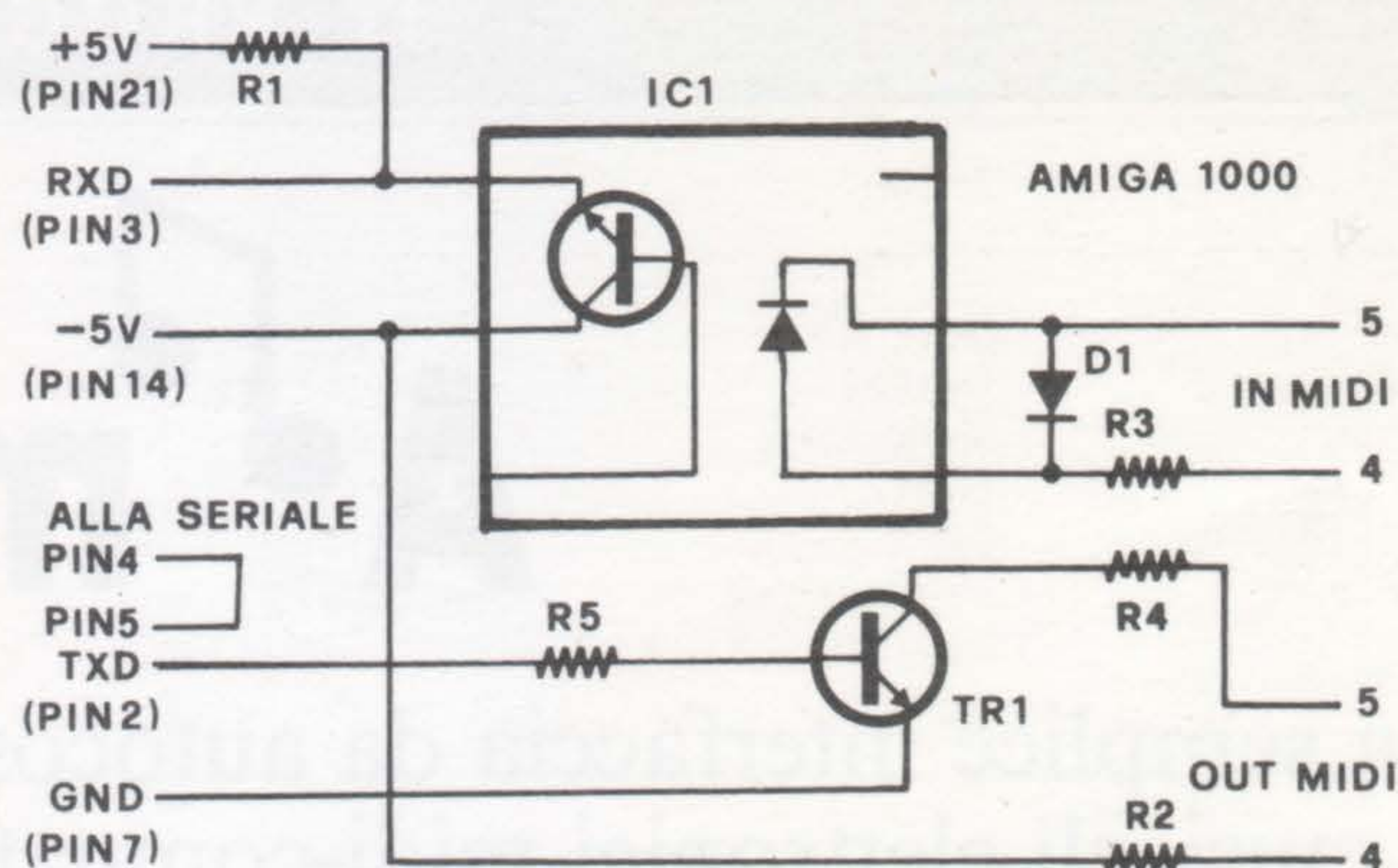
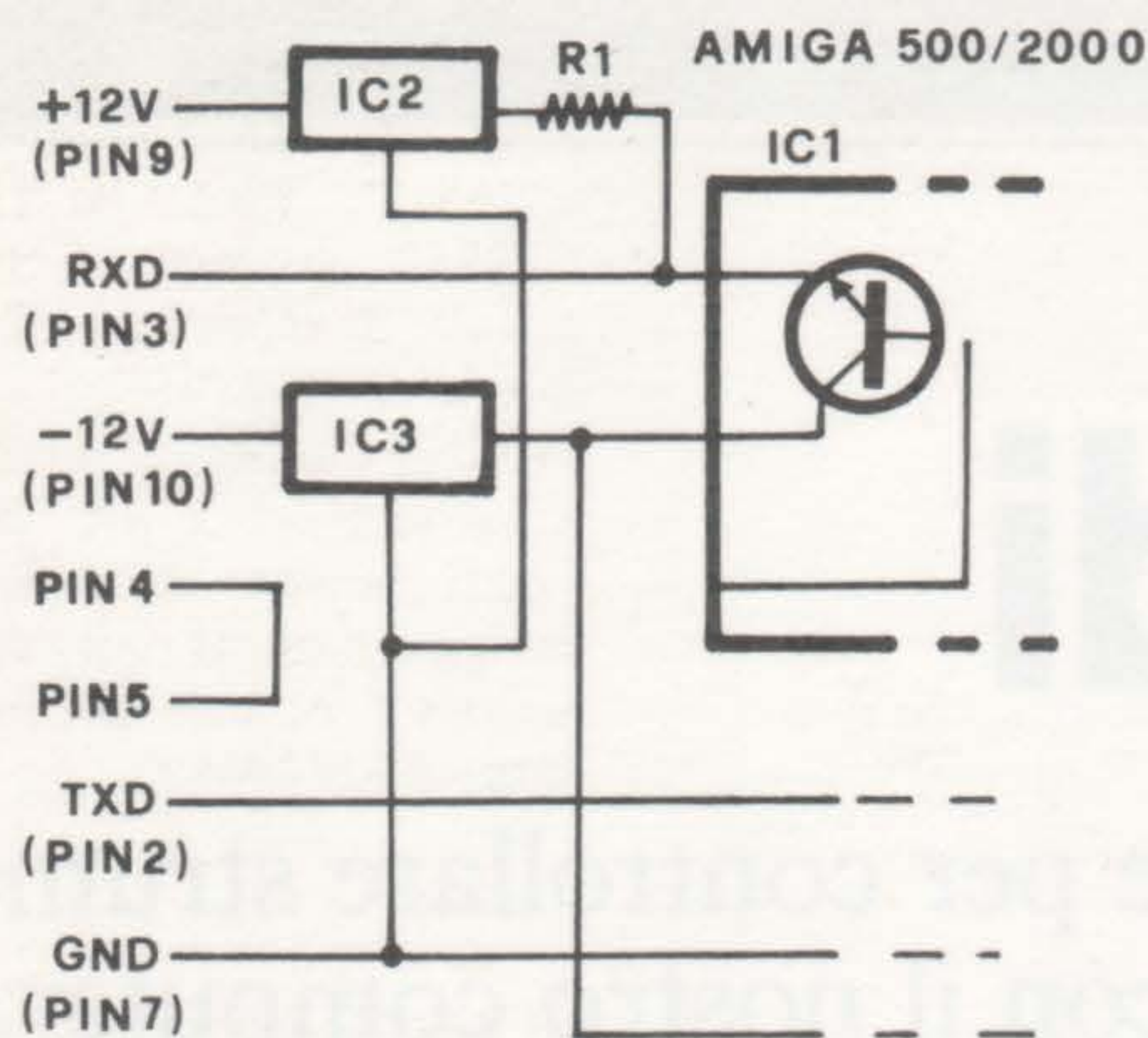
Le sue capacità dipendono totalmente dal genere di interfaccia e dal tipo di sintetizzatori che state usando. In pratica si potranno manipolare fino a 16 canali indipendenti, che potranno così garantire il controllo di altrettanti strumenti totalmente diversi, ognuno con la propria partitura.

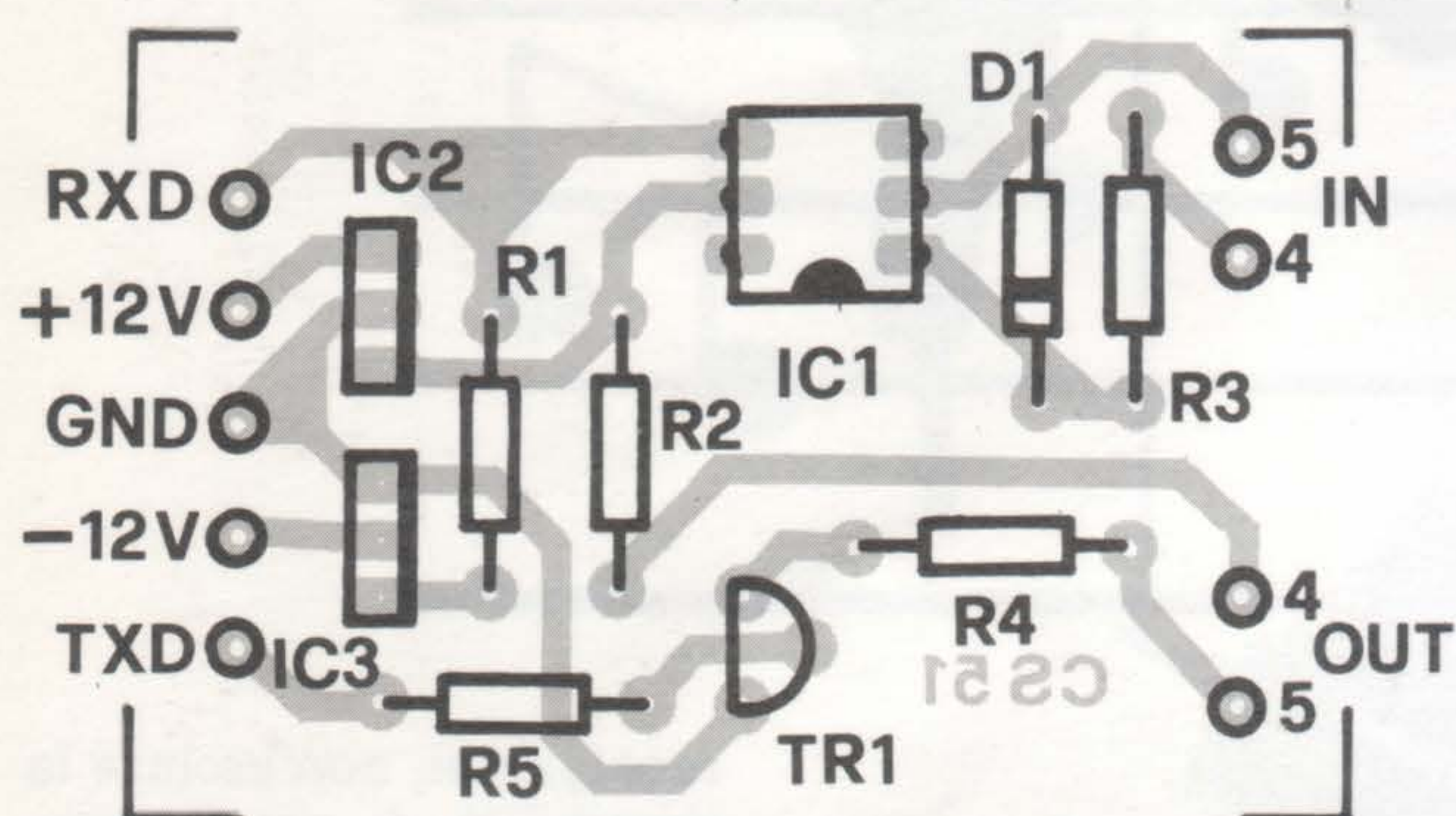
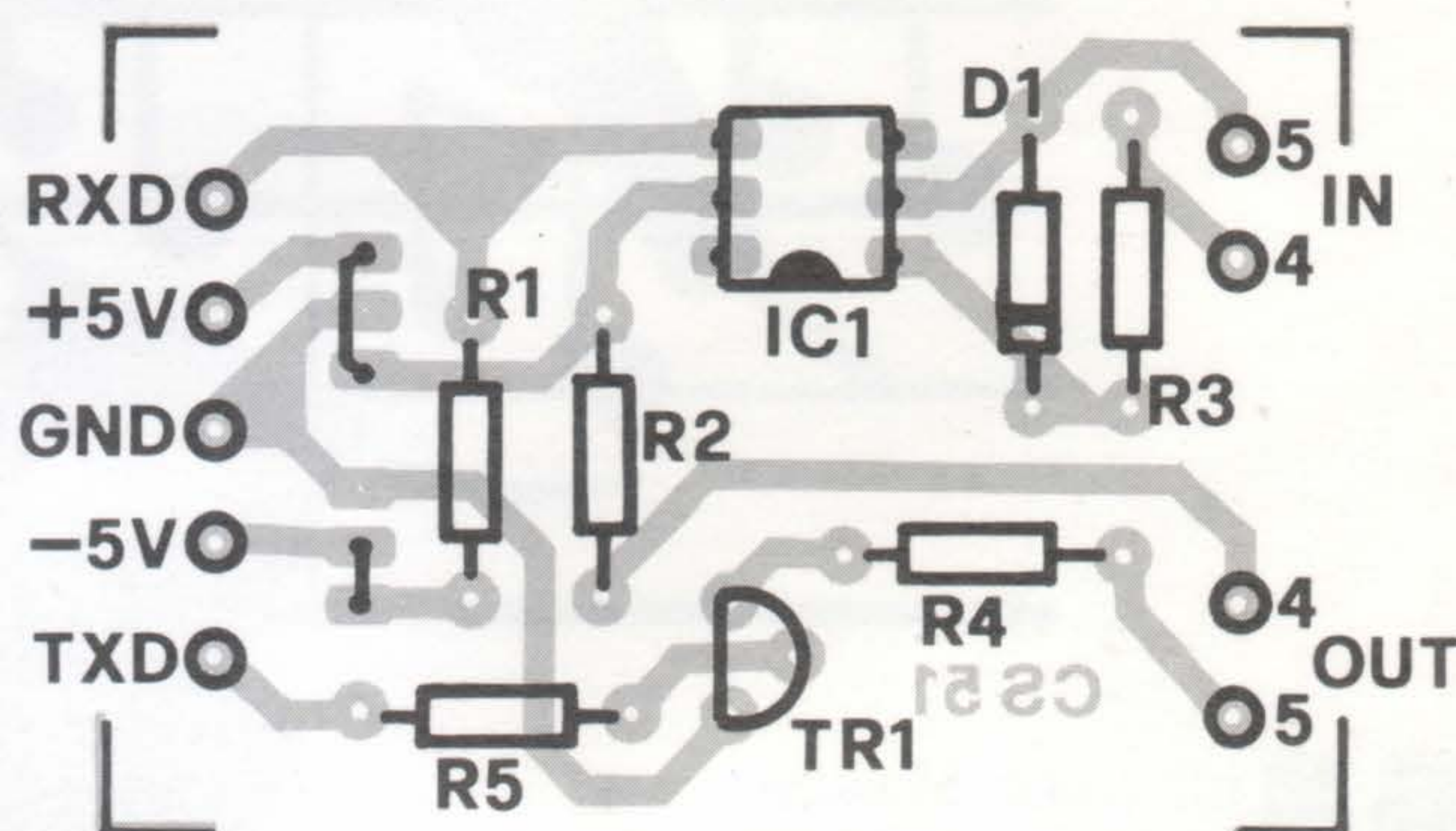
L'idea di questa interfaccia è scaturita sfogliando il Kernel di Amiga che, proponendo tutti gli svariati parametri di settaggio delle periferiche, fa riferimento alla trasmissione ad alta velocità. Per poter trasmettere in questo formato bisogna infatti ricevere e trasmettere alla velocità di 31250 baud, velocità quasi straordinaria per una seriale di tipo convenzionale. Lo stupore aumenta nell'apprendere che, settando un solo flag (per l'esattezza il SERB_RAD_BOOGIE), si può automaticamente aumentare al massimo la velocità della seriale, disabilitare il controllo di parità, escludere lo XON/XOFF, usare solo parole di 8 bit e cancellare il controllo dei break! Ancor più sconcertante è capire che tutti questi armeggi vengono fatti sì per rispettare le regole dello standard, ma soprattutto per poter reggere la velocità!!!

Proprio così, Amiga sembra sia al limite delle sue possibilità. La conferma si ha leggendo due righe più in basso, laddove viene detto che, per non rischiare di fallire la trasmissione o la ricezione, è vivamente consigliato di non usare troppo tempo di CPU. Comunque, visti i programmi che circolano, si può affermare che è possibile fare veramente parecchio (in qualche caso anche troppo, data la complessità di certi software). Già il vecchio, ma sempre valido, Deluxe Music Construction Set ha delle caratteristiche molto professionali: ha tutte le possibili regolazioni, non esclusa la possibilità di usarlo proprio come sequencer.

Nonostante questi programmi e certe attrezzature siano per uso professionale, non è da escludere che qualche amante della musica fatta in casa possa trovare un valido aiuto anche in queste pagine, più che altro esemplificative.

A scopo di riferimento è utile segnalare quali dovrebbero essere le caratteristiche di un'interfaccia completa: la MIDI dovrebbe sia trasmettere che ricevere, oltre che le note: il cambio dei programmi, i timbri, la posizione del PITCH BENDER, la profondità di modulazione e, per finire, anche la velocità di pressione del tasto.

Tutto questo è subordinato al programma, all'interfaccia ed agli strumenti impiegati: basta che uno di questi sgarri per compromettere il risultato finale.

Schema elettrico del circuito: a sinistra la versione 500/2000, a destra la 1000.

VERSIONE PER 500/2000

VERSIONE PER 1000

Potrebbero interessare alcuni codici, per esempio quelli di una batteria elettronica. (Ogni dato completo è formato da 3 byte per quanto riguarda le note, 2 byte per il selettore di traccia e solo 1 byte per i segnali di sincronizzazione). Si veda la tabella. Dove c'è logicamente solo una parte della lunga lista che, una volta trasmessa, dice cosa fare alla batteria elettronica; oltre a questa lista ci sono i codici di ricezione, lievemente diversi ma strutturalmente identici.

BASTA SOLO COLLEGARE?

Ora, il problema sembra essere solamente quello di collegare l'Amiga ai nostri strumenti musicali. Teoricamente, visto che la seriale «pensa lei» alla velocità, i programmi provvedono a

gestire i segnali che arrivano ed i livelli logici sono quelli che ci servono, potrebbe bastare attaccare i quattro fili interessati.

Il discorso non fà una grinza, salvo per il fatto che lo standard è poco standard, cioè non è possibile sapere a priori che tensioni trasmettano gli strumenti musicali che attaccheremo ad Amiga.

Infatti, se il voltaggio presente ai capi dell'8520 (la nostra seriale) dovesse raggiungere una soglia critica, potrebbe succedere un guaio. Così, basterà disaccoppiare l'entrata e, visto che ci siamo, amplificare un po' l'uscita. Guardando lo schema elettrico, noterete infatti che in entrata il segnale passa attraverso una resistenza (R3) che lo riduce in ampiezza per il fotodiodo interno al 4N27 (la sua tensione di lavoro è di circa un volt, un volt e mezzo,

mentre l'entrata si aggira, quando va bene, sui 5 o 6 volt), passa poi ad un diodo 1N4148, usato in caso di tensioni inverse troppo insidiose e, per finire, arriva all'accoppiatore ottico. Questo componente serve appunto in quei casi in cui l'isolamento è necessario (alcuni modelli arrivano a isolare ben 6000 volt, sì, proprio seimila!) ma la trasmissione del segnale deve rimanere integra durante il passaggio.

CON IL FOTOTRANSISTOR

Il fotodiodo interno emette luce in modo che il fototransistor che gli sta a fianco riesca a far passare i 5 volt che sono presenti sul suo collettore all'emettitore. Quest'ultimo è stato polarizzato preventivamente con una tensione nega-

Tabella MIDI

SEZIONE NOTE

byte stato	secondo	terzo
1001 nnnn (n canale)	0kkk kkkk (nota)	0000 0000 (disabilita nota)
1001 nnnn (n canale)	0kkk kkkk (nota)	0vvv vvvv (accento)
1111 0010 (puntatore nota)	0xxx xxxx (byte meno significativo)	0yyy yyyy (byte piu' significativo)

SELETTORE TRACCIA

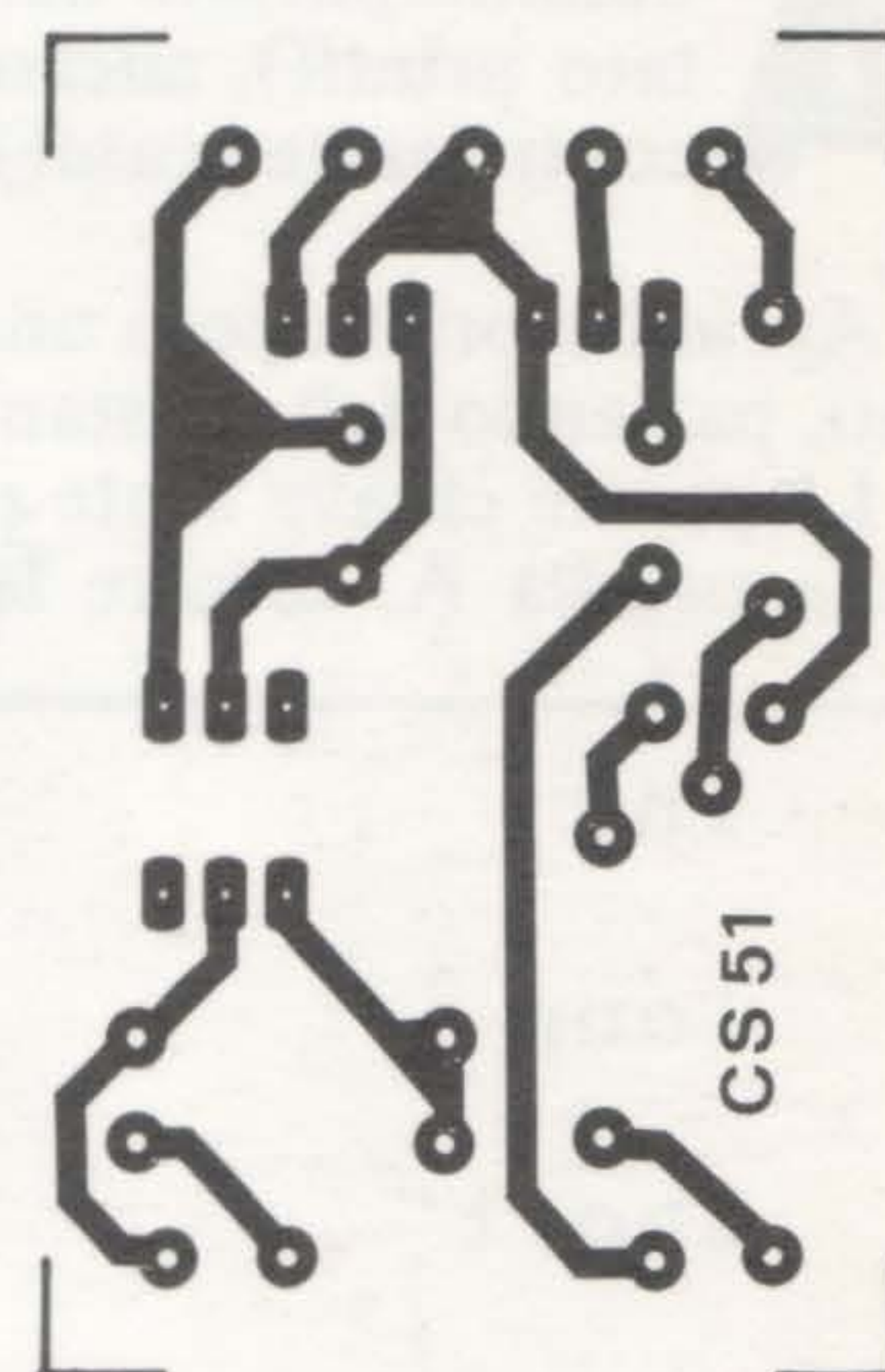
1111 0011	0sss ssss (n traccia)	

SINCRONIZZAZIONI

1111 1010 (impulso di start)
1111 1100 (impulso di stop)

Tabella riassuntiva dei codici Midi: un dato completo è formato da 3 byte.

COMPONENTI: R1= 2200 ohm, R2=220 ohm, R3=220 ohm, R4=220 ohm, R5=10 Kohm, D1= 1N4148, TR1=BC239 o BC547 o similari, IC1= 4N27 o CNY17, *IC2= 7805, *IC3=7905, DB25 maschio o *femmina, 2 connettori 5 pin in 180 gradi da pannello. (I componenti segnati con l'asterisco servono solo per la versione 500/2000).

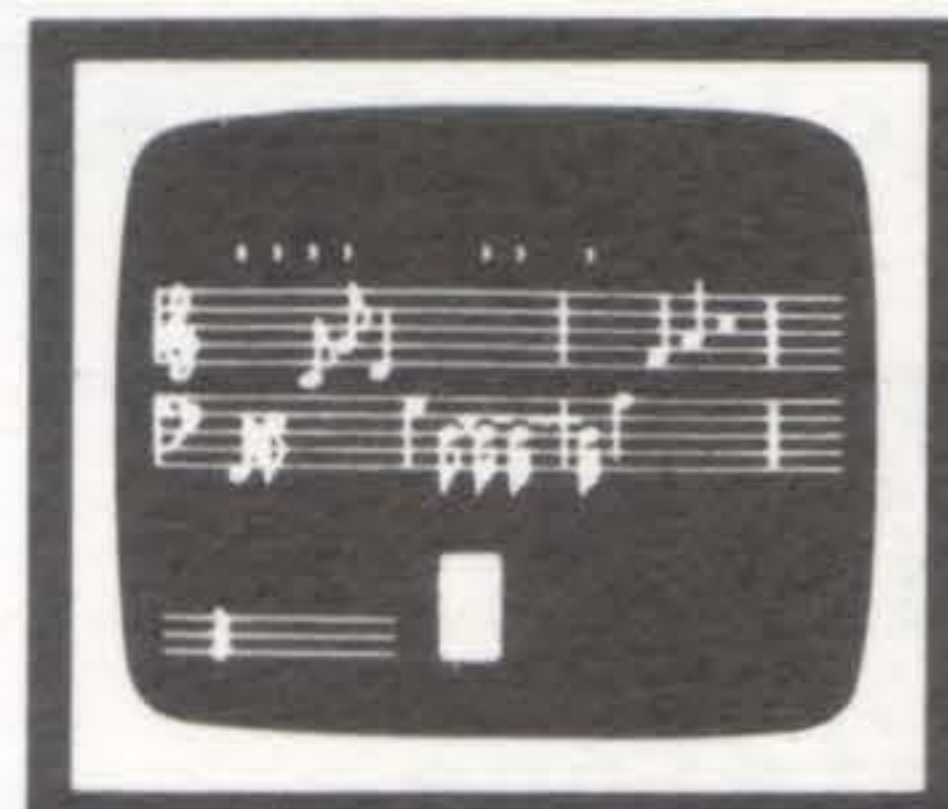
Traccia rame della basetta, che può essere usata, con i dovuti accorgimenti, per entrambe le versioni.

SE PROPRIO VUOI UN'INTERFACCIA PRONTA

Se vuoi un'interfaccia midi perfettamente costruita e già collaudata, completa di IN, OUT, THRU, connettori da 7 poli che possono accettare sia i normali a 5 poli sia una futura espansioncina per moltiplicare gli ingressi e le uscite, invia un vaglia postale ordinario di Lire 55.000, specificando per quale modello Amiga (500/1000/2000) la vuoi, intestato a Mobysoft, c/o Arcadia srl, corso Vitt. Emanuele 15, 20122 Milano.

tiva (attraverso R1) che serve a settarlo, in condizione di mancanza di segnale, a livello logico 0. Questo terminale viene collegato all'RXD della seriale che provvede a riceverlo.

La seconda parte del progetto riguarda l'OUT, ovvero la trasmissione dal computer agli strumenti. Il segnale proveniente dal TXD della seriale giunge attraverso R5 (sempre necessaria per la polarizzazione corretta di TR1) al transistor, che con R4 ed R2 amplifica il segnale pervenutogli.

Il discorso potrebbe concludersi qui, ma sarebbe valido solo per Amiga 1000 nel caso del quale, sull'interfaccia seriale, sono presenti le due tensioni che ci necessitano +5 e —5 volt. Sugli altri due modelli della nostra macchina, invece, i voltaggi sono +12 volt e —12 volt. Basterà comunque ridurre le tensioni con dei 7805 e 7905 messi opportunamente. Va sempre ricordato di guardare bene qual è lo schema del vostro modello di Amiga, perché sbagliando le connessioni si possono arrecare gravi danni al computer.

Per quanto riguarda il montaggio, le possibilità sono due: usare una basetta sperimentale, oppure approfittare del circuito stampato proposto in queste pagine. La prima soluzione risulta pratica e di veloce esecuzione ma resta

bruttina a vedersi; la seconda è senz'altro più ordinata.

Adottando la prima soluzione vi conviene montare i componenti seguendo le indicazioni dei nostri disegni di montaggio; in seguito, con del filo di rame isolato, collegate piedino a piedino seguendo lo schema elettrico (state attenti a non fare falsi contatti o cortocircuiti pericolosi).

Per chi opta per la soluzione dello stampato il lavoro di montaggio sarà più sbrigativo, basterà seguire i disegni di montaggio riguardanti la versione per il 1000 o per 500/2000, saldare tutti i componenti al loro posto nella posizione corretta (attenti alla piedinatura del transistor, a quella dell'accoppiatore ed al diodo) ed il gioco è fatto. Terminato l'assemblaggio della piastra, rimarranno da collegare i connettori di IN e di out ed il BD25; controllate, sul corpo plastico, la piedinatura ed il numero, stando attenti a non invertire nulla. Tutto qui.

La prova del nove avverrà collegando l'interfaccia all'Amiga e ad uno strumento musicale (batteria, synt, tastiera) e facendo partire un programma musicale, provando se tutto funziona correttamente. ■

"C"

LE VARIABILI DEL C

2ª puntata

di ALESSANDRO BOSSI

Abbiamo parlato della funzione di output formattato printf(), adesso diamo un'occhiata alla sua controparte scanf(), funzione di input formattato.

Apriamo prima però una parentesi sui dati ed i tipi di dati, parlando delle costanti e delle variabili.

Le parole chiave usate per i tipi di dati sono indicate nella tabella A, mentre le loro dimensioni ed i valori minimi e massimi sono indicati nella tabella B.

int	intero
long	lungo
short	corto
unsigned	senza segno
char	carattere
float	virgola mobile
double	virgola mobile in doppia precisione

Tabella A. Le parole chiave usate per i tipi di dati.

Nel caso di una costante, il compilatore è in grado di riconoscere automaticamente di quale tipo si tratti, ma nel caso delle variabili dobbiamo dichiarare quali variabili intendiamo utilizzare nel programma ed a quale tipo appartengono. Prendiamo ad esempio questo breve programma, contenuto nel dischetto allegato ad Amiga Byte n. 2 sotto il nome «prog. 1»:

```
main()
{
int prova;
prova = 12345;
printf("La variabile prova vale: %d\n",prova);
}
```

Le variabili vengono dichiarate subito dopo l'apertura della parentesi graffa che segnala l'inizio della funzione. In questo caso abbiamo dichiarato la variabile prova, di tipo intero, quindi il compilatore riserva lo spazio necessario a contenerla (32 bit). In seguito abbiamo assegnato il valore 12345 alla variabile, e quindi abbiamo utilizzato la funzione printf(), usando %d nella stringa di controllo ed indicando prova come argomento. Proviamo a scrivere un programma interattivo e con più variabili («prog. 2» nel nostro dischetto):

```
void main()
{
int num1, num2, somma, diff;
printf("Inserisci il primo numero: ");
scanf("%d", &num1);
printf("Inserisci il secondo numero: ");
scanf("%d", &num2);
somma = num1+num2;
diff = num1—num2;
printf("Somma: %d, differenza: %d\n",somma,diff);
}
```

Spiegheremo più avanti il significato dell'operatore &.

Scrivendo in maniera più compatta avremmo potuto risparmiare fatica e spazio in memoria («prog. 3» del nostro dischetto):

```
void main()
{
int num1, num2;
printf("Inserisci il primo numero: ");
scanf("%d", &num1);
printf("Inserisci il secondo numero: ");
scanf("%d", &num2);
printf("Somma: %d, differenza: %d\n",
num1+num2,num1—num2);
}
```

Esaminando questo programma notiamo che nella dichiarazione delle variabili abbiamo usato la parola chiave int solo all'inizio della linea, separando tutti gli argomenti da una virgola. Questo è valido in qualsiasi dichiarazione, quando dobbiamo dichiarare più variabili di uno stesso tipo. Ad esempio:

```
int sum1, sum2;
int prova;
char tasto1, tasto2, tasto3;
```

Possiamo quindi assegnare alle variabili sum1 e sum2 un valore int con segno, compreso fra —2.147.483.648 e

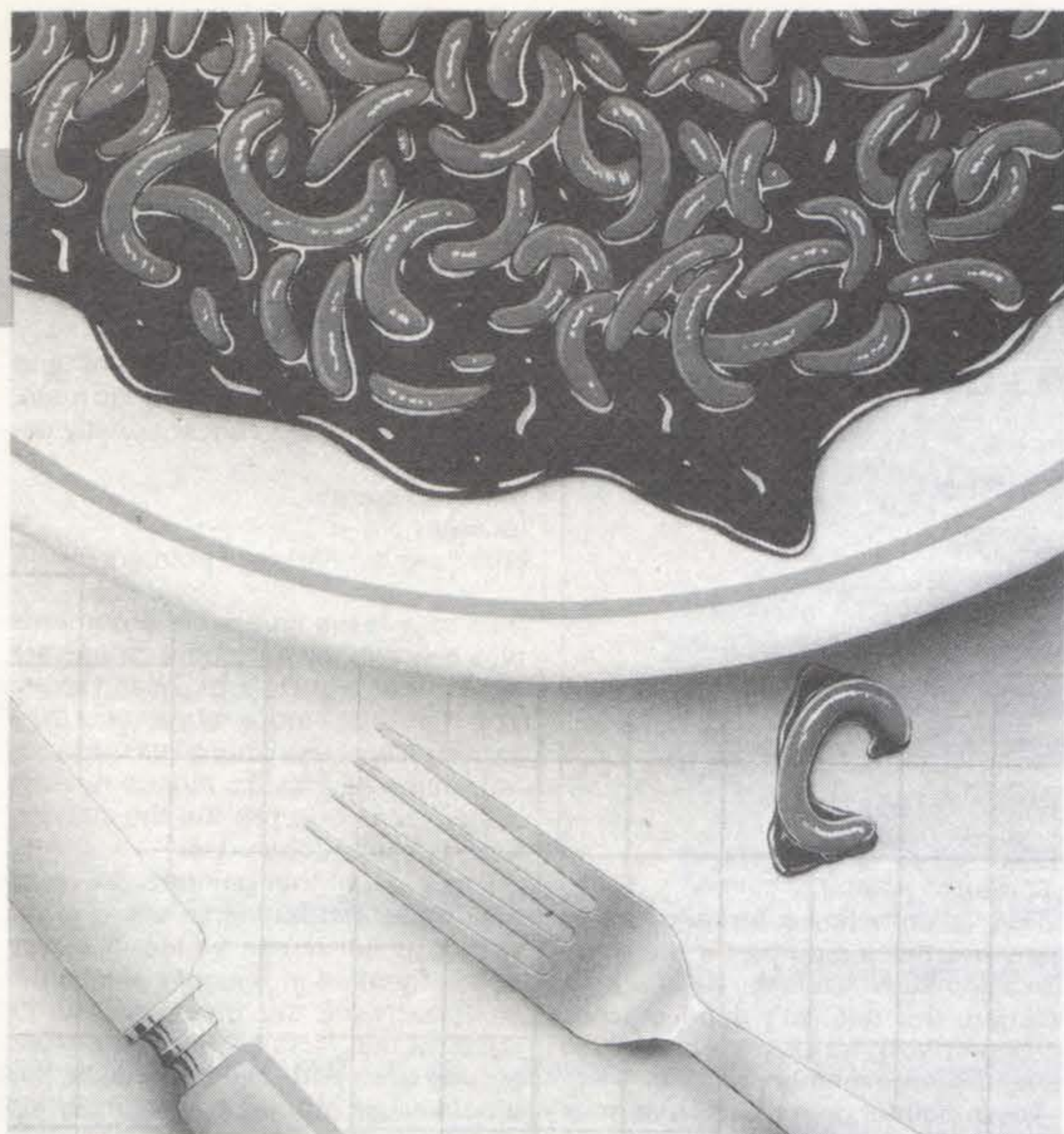



Tabella C

Operatore e significato	Esempio
= assegnazione	num1 = num2
+ addizione	num1 + num2
- sottrazione	num1 - num2
- segno	num1 = -num2
* moltiplicazione	num1 * num2
/ divisione	num1 / num2
% modulo	num1 % num2

2)

Tabella C. Operazioni aritmetiche fondamentali e loro significato.

+2.147.483.647. Attenzione, però, alla somma.

Se ciascuna delle due variabili contiene, per esempio, +2.000.000.000, sommandole insieme otteniamo un valore che sfonda il tetto massimo consentito alla variabile int, andando incontro ad un errore ed alla conseguente interruzione del programma.

LE ALTRE OPERAZIONI

Oltre alle operazioni di somma e di sottrazione, esistono naturalmente anche le altre operazioni fondamentali, indicate nella tabella C, con le loro priorità fornite nella tabella D. L'unico operatore che necessita di ulteriori spiegazioni è quello di modulo. Questo operatore restituisce il resto della divisione intera fra i due operandi. In pratica, se abbiamo 21 % 4, l'operatore divide 21 per 4, con risultato 5, poi moltiplica 5 per 4, con risultato 20, quindi sottrae 20 da 21 ottenendo come risultato finale 1. Questa operazione può essere utile in molti casi; per vedere, ad esempio, se un numero è divisibile per un altro.

Se il risultato dell'operazione modulo è zero, allora il primo operando è un multiplo del secondo; vale a dire che è perfettamente divisibile per quest'ultimo, senza resto.

Per ottenere il risultato intero della divisione sopra citata usiamo invece l'operatore di divisione /, che ci restituisce il risultato della divisione, troncando l'eventuale parte decimale (in questo caso ci restituisce 5). Questo vale però solo per operazioni sugli interi.

In generale gli operatori si adattano al tipo di dati sui quali stanno lavorando, fornendo il risultato nel formato degli argomenti dell'operazione. Se scriviamo 15 / 6 otterremo 2, ma se scriviamo 15.0 / 6 oppure 15 / 6.0

TIPO	BITS	MINIMO	MASSIMO
char	8	-128	+127
unsigned char	8	0	255
short int	16	-32768	+32767
unsigned short int	16	0	65535
int	32	-2.147.483.648	+2.147.483.647
unsigned int	32	0	4.294.987.295
long int	32	-2.147.483.648	+2.147.483.647
unsigned long int	32	0	4.294.987.295
float	32	+/-10E-37	+/-10E+38
double	64	+/-10E-37	+/-10E+38
pointer (*)	32	0	0xFFFFFFFF

Tabella B. Dimensioni e valori minimi e massimi dei tipi di dati.

Operatori	Associativita'
( )	da sinistra a destra
- (segno)	da sinistra a destra
* / %	da sinistra a destra
+ - (sottrazione)	da sinistra a destra
=	da destra a sinistra

Tabella D. Priorità degli operatori aritmetici fondamentali in ordine decrescente e loro associatività.

oppure 15.0 / 6.0 otterremo 2.5. Che cosa c'è di diverso nelle due scritture? Nel primo caso dividiamo due interi, quindi il risultato è un intero, mentre nel quarto caso i due argomenti sono in virgola mobile, e tale è anche il risultato. Nel secondo caso, invece, dividiamo un numero in virgola mobile (15.0) per un intero (6) e nel terzo caso un intero (15) per uno in virgola mobile (6.0). Questo fa sì che il compilatore scelga, come rappresentazione del risultato finale, quella scrittura che offre la precisione maggiore fra le due, in questo caso quella in virgola mobile.

Continuiamo a parlare di tipi di dati e diamo un'occhiata al tipo char (carattere). Le variabili e le costanti char sono rappresentate da un numero ad 8 bit, che generalmente rappresenta il codice ASCII di un carattere.

Un programma di esempio («prog. 4» sul nostro dischetto):

```
void main()
{
char tasto;
printf("Premi un tasto (e poi RETURN) ");
scanf("%c", &tasto);
printf("\nHai premuto %c\n",tasto);
}
```

Questo banale programmino attende che l'utente prema un tasto (e segnali poi di avere terminato l'immissione con il tasto return), e poi ristampa il tasto precedentemente premuto. Come possiamo fare per inserire e ristampare una stringa di caratteri? Per questa operazione è necessario introdurre gli array. Gli array sono costituiti da una serie di variabili dello stesso tipo, che portano lo stesso nome, munito di un indice numerico che identifica ciascun elemento in maniera biunivoca. Ad esempio:

char nome[30];

è la dichiarazione di un array di 30 elementi di tipo char.

Le dichiarazioni si leggono da destra verso sinistra, quindi la prima cosa che incontriamo sulla linea è [30] che ci dice «array di 30 elementi»; poi troviamo il nome della variabile, quindi il tipo di dati che costituisce l'array.

Il programma («prog. 5» del nostro dischetto):

```
void main()
{
char nome[10], cogn[10];
int anni;
printf("Nome? ");
scanf("%s", &nome);
printf("Cognome? ");
scanf("%s", &cogn);
printf("Anni ? ");
scanf("%d", &anni);
printf("\nDati anagrafici: %s %s, anni %d\n",
nome,cogn,anni);
}
```

chiede-accetta come input, alla prima richiesta, una stringa alfanumerica di 30 caratteri al massimo (29, visto che uno serve come segnalatore di fine stringa); di 35 (34) nel secondo caso; un carattere alfanumerico nel terzo caso ed un numero compreso fra —32768 e +32767 nel quarto.

Le stringhe, quindi, vengono trattate come array di caratteri, nei quali potete accedere ad un elemento qualunque.

Rispondiamo a questo programma con:

Nome? Marco
Cognome? Cassinari
Anni? 24

Nella memoria interna del calcolatore i nostri dati sono immagazzinati come nella tabella E.

Potremmo chiedere la stampa di uno qualsiasi degli elementi dell'array. Ad esempio:

printf("elemento 3 di nome: %c",nome[3]);

stamperebbe la lettera «c». (Quelli che pensavano alla lettera «r» si tirino un orecchio!).

Nella dichiarazione, indichiamo il numero massimo degli elementi che l'array può contenere (in questo caso 10), ma poi gli elementi vengono indirizzati a partire dall'elemento [0] che è il primo dell'array, mentre [1] è il secondo e così via, fino a [9] che è l'ultimo.

Se tentassimo di scrivere:

printf("elemento 10 di nome: %c",nome[10]);

otterremmo un bel messaggio di errore dal compilatore.

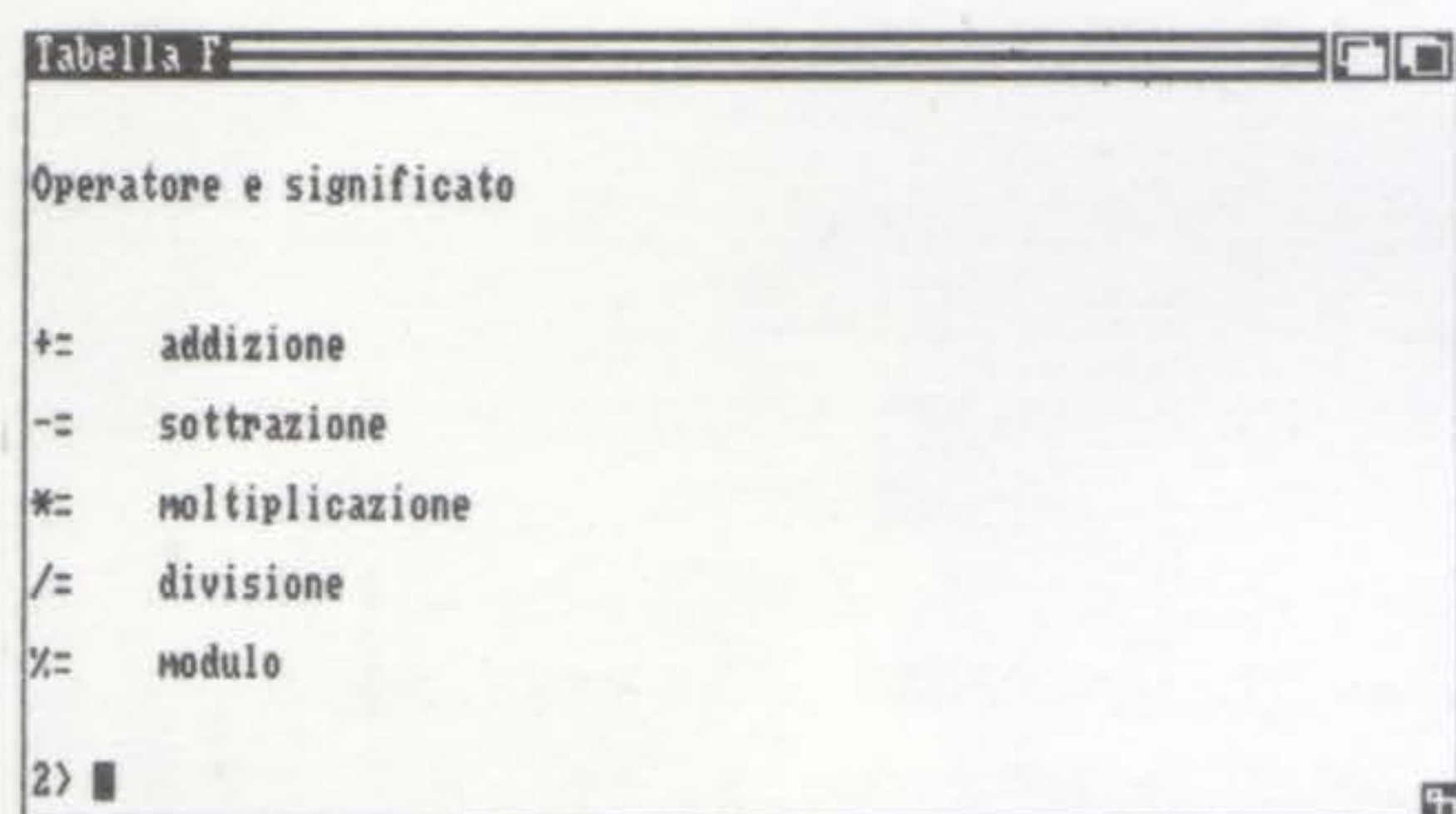
L'elemento 3 è quindi il quarto dell'array, cioè la lettera «c». Per avere la possibilità di scrivere programmi un pochino più complessi, apriamo una parentesi sul ciclo FOR.

La sintassi del ciclo for è:

```
for (esp1; esp2; esp3)
{
. . . .
}
```

dove esp1, esp2, esp3 sono tre espressioni di controllo, che controllano appunto l'iterazione nel seguente modo:

elemento	0	1	2	3	4	5	6	7	8	9
char nome[10] =	'M'	'a'	'r'	'c'	'o'	0	x	x	x	x
char cogn[10] =	'C'	'a'	's'	's'	'i'	'n'	'a'	'r'	'i'	0
int anni =	24									

Tabella E. Rappresentazione in memoria dei dati generati da Prog. 5.

1) L'espressione esp1, detta di inizializzazione, viene eseguita una volta sola, all'entrata del ciclo.
2) Se l'espressione esp2, detta «condizione», è vera (diversa da zero), vengono eseguite le istruzioni racchiuse fra le parentesi graffe; altrimenti (falsa, 0), si salta all'istruzione che segue la coppia di parentesi graffe, interrompendo il ciclo.
3) Dopo aver eseguito l'ultima istruzione racchiusa fra le parentesi graffe, viene calcolata l'espressione esp3, detta di aggiornamento, quindi si torna al punto 2.

In pratica, la parentesi graffa chiusa è equivalente all'istruzione NEXT che chiude un ciclo FOR in Basic.

Proviamo questo programma («prog. 6» del nostro dischetto):

```
void main()
{
int n;
char prova[10];
printf("Inserisci una stringa di 9 caratteri: ");
scanf("%s", &prova);
printf("\nHai inserito: %s/n",prova);
for (n=0; n < 10; n=n+1)
  {
  printf("elemento %d: %c, ASCII: %d",n,prova[n],
  prova[n]);
  printf("\n");
  }
}
```

Tabella F

Operatore e significato

+=	addizione
-=	sottrazione
*=	moltiplicazione
/=	divisione
%=	modulo

2>

Tabella F. Elenco degli operatori di assegnamento addizionali.

Inseriamo una stringa di caratteri alfanumerici a piacere, e vedremo la stringa stampata elemento per elemento, con il valore ASCII accanto. Troverete una tabella ASCII in appendice al manuale dell'Amiga Basic che avete ricevuto insieme al computer.

Ma esaminiamo da vicino il programma. Viene letta una stringa di 9 caratteri (o meno) e viene messa in un array di 10 elementi di cui l'ultimo, uno zero, indica la fine della stringa per le funzioni come printf(). Se terminate la stringa prima di 9 caratteri, lo zero verrà posto immediatamente dopo l'ultimo carattere inserito, ed il resto dell'array conterrà valori casuali.

Il ciclo for inizia con l'espressione n=0, che assegna zero alla variabile n, che useremo come «contatore» del ciclo. Quindi si verifica che n sia minore di nove, condizione verificata, e si eseguono i due printf() all'interno delle graffe.

La prima printf() stampa il valore del «contatore», o numero progressivo dell'elemento dell'array, quindi stampa il carattere dell'array ed il suo valore ASCII.

Come possiamo notare, gli ultimi due argomenti di printf() sono gli stessi. Questo si spiega con il fatto che i caratteri, all'interno della macchina, sono rappresentati da numeri interi, compresi fra 0 e 255, la cosiddetta codifica ASCII. Noi chiediamo a printf() di stampare un carattere (%c) e gli diamo come argomento un elemento di un array di caratteri, internamente rappresentato da un array di interi ASCII, che viene quindi convertito in carattere e stampato. Poi chiediamo a printf() di prendere lo stesso elemento e di stamparne direttamente il valore ASCII.

La seconda printf() è utilizzata per andare a capo dopo ogni elemento. Proviamo adesso ad ottimizzare questo programma, scrivendo il codice sorgente in maniera più compatta, quindi riducendo l'eseguibile («prog. 7» del nostro dischetto):

```
void main()
{
int n;
char prova[10];
printf("Inserisci una stringa di 9 caratteri: ");
scanf("%s", &prova);
printf("\nHai inserito: %s\n",prova);
for (n=0; n < 10; n+=1)
  {
  printf("elemento %d: %c, ASCII: %d\n",n,prova[n],
  prova[n]);
  }
}
```

Nell'espressione di aggiornamento del ciclo for, abbiamo sostituito n=n+1 con una notazione analoga ma più breve. Questa serie di operatori di assegnamento è elencata nella tabella F.

Per incrementare o decrementare di un'unità una variabile, esistono due appositi operatori, detti di incremento e decremento: ++ e ——.

Nel nostro caso possiamo quindi scrivere:

```
for (n=0; n < 10; n++)
```

A seconda delle esigenze, l'operatore di incremento ++, può essere preposto o postposto ad una variabile o ad una espressione, ed agisce sul valore di quest'ultima, prima o dopo che il suo valore venga utilizzato nella espressione che la contiene. Nel nostro caso è indifferente scrivere n++ o ++n, in quanto la variabile compare da sola, e non in un'espressione.

Possiamo anche risparmiarci la scrittura delle due parentesi del ciclo for, in quanto il corpo del ciclo è composto da una sola istruzione. In questo caso, e solo in questo, è possibile omettere le parentesi graffe.

Il ciclo for diventerà quindi:

```
for (n=0; n < 10; n++)
printf("elemento %d: %c, ASCII: %d\n",n,prova[n],
prova[n]);
```

L'ultima parola di questa puntata va dedicata allo «stile». Dal primo esempio con il ciclo for, abbiamo iniziato a spostare alcune linee di istruzione verso destra di 2-3 caratteri. Questa tecnica, detta di «indentare», serve a mettere visivamente in risalto i blocchi logici del programma, a facilitarne la lettura. Anche se impareremo in seguito ad apprezzare questa tecnica, è importante abituarsi ad utilizzarla da adesso, per gli indiscutibili vantaggi che comporta.

Ogni coppia di parentesi graffe (aperta e chiusa) dovrebbe trovarsi sulla stessa colonna, e così il codice contenuto all'interno dovrebbe essere allineato su questa colonna. Qualora si dovessero avere più cicli o espressioni condizionali, costruiti l'uno nell'altro, si dovrebbe indentare ulteriormente di 2-3 caratteri all'inizio di ciascun blocco logico (insieme di istruzioni costituenti il ciclo o l'espressione condizionale).

■

MISTER KIT
Elettronica 2000

ELETTRONICA APPLICATA, SCIENZA E TECNICA

From Los Angeles

INTERFACCE A GOGO

L'americana CSA ha presentato ufficialmente la nuova scheda basata sul microprocessore 68030 coadiuvato dal 68882, denominata Over 30 Board. Riesce a girare 4 volte più veloce della scheda sorella con il 68020/68881 ed i primi test danno una performance di circa 8 Mips (milioni di operazioni al secondo).

Altre interfacce interessanti sono la Bernoulli Drive and Interface (della Inner Connection) ed il Panasonic WORM (Write Once Read Many, cioè scrivi una volta ma leggi sempre): ogni dischetto (simile ai normali compact disc audio) può contenere 400 Mega Byte (!) con un accesso medio di 2.5 MB al secondo. Nuovo Word Processor della Brown-Wagh denominato Excellence!, che dovrebbe avere tutto l'immaginabile: spelling, grafica IFF, colore, supporto del Postscript e tantissime altre cose. Un'altra ciliegina software riguarda la telematica ed

un relativo nuovo pacchetto che, a detta della pubblicità, annovera tutte le opzioni possibili: il suo nome è AmicTerm e lo recensiremo al più presto.

TOOLKIT

Disk Utilities

Ulteriore aggiornamento del toolkit Diskwick (Tigress, PO Box 665, Glendora, CA 91740) giunto alla 2.0: aggiunte opzioni quali un nuovo editor per modificare i settori, ricerca di stringhe ascii o esadecimali, hard copy di blocchi, inseguimento dei blocchi di un determinato programma, checksum dei boot block e dischi Kickstart.

La Mirror Soft annuncia che non ci saranno più problemi con dischi danneggiati usando il suo File Rescue Plus, in grado anche di recuperare eventuali file cancellati, ovviamente entro determinati limiti. Una delle peculiarità è data dall'abilità di lavorare anche con dischi formattati non AmigaDos, come accade con alcuni programmi di word processor o spread sheat: particolare, questo, da non sottovalutare.

ROBTEK GAMES

CHAMP: nuovo boxing game pubblicizzato come il più realistico mai visto fin'ora. CRAZY FOOTBALL: arcade game caratterizzato da quattro palle che devono essere portate fuori contemporaneamente da un labirinto. PINBALL: conversione del primo e famosissimo flipper americano. STARWAYS: molto simile al gettonatissimo Trailblazer, con la differenza che non guiderai più la pallina ma la pista!

Nuove release

DESKTOP PUBLISHING

Nuova release Soft-Logic (11137 South Towne Sq. Ste. C, St.

Louis, MO 63123) del proprio programma di DTP, chiamato ora Publishing Partner Professional: parecchie le variazioni, che vanno dal supporto completo del colore, di tutti gli standard grafici (IFF, TIFF, HAM) all'auto-hypenation ed alla possibilità di ruotare testi/oggetti direttamente sul video. Stampa compatibile con la maggior parte delle stampanti a 9, 18 e 24 aghi, oltre che con le famose Hewlett Packard Laserjet.

SUL DISCHETTO

Il dischetto di questo mese contiene sei programmi tutti di pubblico dominio, scelti con cura per soddisfare le esigenze più diverse. Ogni programma è accompagnato da dettagliate istruzioni in italiano ed in inglese. Ecco qui di seguito un breve sommario.

GUARDIAN

Novità assoluta per l'Italia, è in grado di debellare per sempre il famigerato virus ed è addirittura capace di individuare i dischetti sospetti e di «curare» quelli infetti.

MATHFONT

In esclusiva per Amiga Byte, un set di caratteri matematico utilizzabile nei propri programmi e richiamabile da qualsiasi word-processor o programma di grafica che permetta il caricamento di font esterni.

DIRMASTER

Devi riordinare e catalogare il tuo archivio dischetti in modo razionale? Ecco un programma potentissimo che consente di memorizzare e riorganizzare migliaia di titoli.

STARCHART

Dedicato a tutti gli appassionati di astronomia, visualizza una mappa completa del cielo e fornisce tutte le informazioni su pianeti, astri e costellazioni.

GOMF

Finalmente risolto l'odioso problema delle Guru Meditation: basta installare Gomf nella memoria di Amiga e provvederà ad intercettare ed eliminare le cause del blocco del sistema.

MASTERMIND

Il classico gioco d'intelligenza in versione computerizzata.
Il giocatore deve indovinare la combinazione di colori pensata da Amiga.

INOLTRE...

... sono presenti, da questo fascicolo in poi, sempre sul dischetto, tutti gli esempi relativi alle puntate contemporanee dei nostri corsi di «C» e di Assembler. Fascicolo alla mano, basterà cercarli sul disco risparmiandosi la fatica di battere i listati.

LE VOSTRE IDEE, I VOSTRI PROGRAMMI

La redazione di Amiga Byte invita tutti quelli di voi che hanno idee o programmi già realizzati ad inviarceli in visione o a proporceli. I programmi debbono essere su dischetto, completi di istruzioni, ed assolutamente originali, ovvero pensati e realizzati da voi. Mandate sempre una copia perché tutto il materiale, pubblicato o meno, non viene restituito. Qualunque programma pubblicato verrà compensato previo accordo con l'autore. Indirizzate i vostri lavori ad Amiga Byte, Arcadia srl, c.so Vitt. Emanuele 15, 20122 Milano.

Cosa s'intende con questo strano termine che evoca brutti quarti d'ora, errori fatali e resettaggio forzoso, e cosa significano le fatidiche 16 cifre del suo codice.

di ENRICO DONNA

Tutti si sono imbattuti, presto o tardi, in quella che i programmatori del sistema operativo Amiga hanno bizzarramente definito «Guru Meditation»: quel brutto quarto d'ora cioè quando, durante l'esecuzione di un programma, il computer incontra un errore fatale al quale è possibile porre rimedio solo resettando la macchina.

L'apparizione della caratteristica finestra con il messaggio

Software error - task held
Finish ALL disk activity
Select CANCEL to reset/debug

equivale purtroppo ad una diagnosi fatale per il programma che state eseguendo in quel momento.

È vero che è possibile ignorare la richiesta e continuare ad usare Amiga anche se il programma in questione è bloccato, ma questo è di poco aiuto quando, ad esempio essendo in corso un'elaborazione di dati, non è più possibile salvare il lavoro svolto fino a quel momento.

Una consolazione è però fornita dal messaggio successivo del sistema, la Guru Meditation vera e propria, che prima di eseguire la sentenza e cancellare i dati in memoria, visualizza sullo schermo un codice di sedici cifre nel quale è spiegata la causa del blocco del sistema.

Questo codice può essere utile per individuare l'errore che ha generato la condizione di blocco, in modo da porvi rimedio per il futuro. Per un programmatore la Guru Meditation può costituire un valido strumento per il debug, e per i normali utilizzatori può essere quantomeno interessante conoscere il motivo per cui il computer ha smesso di funzionare correttamente.

In figura 1 è indicato un esempio di «alert number» (come viene definito il codice visualizzato nella finestra di una Guru Meditation). Le prime due cifre, chiamate «Subsystem Number», indicano in quale modulo del sistema operativo è avvenuto l'errore. Nell'esempio in questione le cifre «07» indicano che si tratta del modulo relativo al DOS.

Se la prima cifra è superiore ad otto, bisogna sottrarre ad essa il numero otto per ottenere il valore reale. Il numero è esadecimale: quindi, se la prima cifra fosse A, il valore reale sarebbe 2. La differenza consiste nel fatto che i codici con la prima cifra uguale o superiore ad 8 indicano, almeno in teoria, errori non definitivi, ai quali è possibile porre rimedio. Nella pratica, tuttavia, si deve quasi sempre ricorrere al reset.

Le due cifre successive, denominate «General error code», sono utili perché consentono di individuare immediatamente la causa del problema, ad esempio la mancanza di memoria o di librerie, o di driver dal disco (il che può accadere quando il disco dal quale è stato effettuato il boot del sistema non ha le directory DEVS o LIBS).

Le quattro cifre che seguono specificano esattamente che errore è avvenuto; la loro corretta interpretazione deve tenere conto anche dei codici precedenti, poiché lo stesso codice, chiamato «Specific error code», può assumere significati diversi in relazione a codici generali di errore differenti. In figura 2 è rappresentata una tabella riassuntiva contenente tutti questi codici di errore e la loro interpretazione, così come è stata dedotta dalla documentazione relativa al Rom Kernel di Amiga.

Le rimanenti otto cifre a destra del punto rappre-

sentano l'indirizzo esadecimale in cui il programma si è interrotto causa la condizione di errore; informazione utilissima per individuare e correggere il codice di un programma in fase di sviluppo.

Tutti questi codici sono memorizzati nel programma Exec/Alert.h, che fà parte del sistema operativo di Amiga. Si possono però verificare condizioni di errore non previste, causate ad esempio da istruzioni non riconosciute dal microprocessore 68000. Quando una di queste condizioni si verifica, il codice di errore visualizzato non segue più le regole precedentemente elencate; le prime sette cifre rimangono azzerate, l'ottava indica una specifica condizione di errore. L'elenco di queste «CPU traps», come vengono chiamate, è riassunto in figura 3.

In ogni caso, prevenire è meglio che curare quindi, seguendo alcune semplici regole di comportamento, è spesso possibile evitare di incorrere nell'inconveniente di una «Guru Meditation».

MEMORY PROBLEM

Ricordate che sovente è la mancanza di memoria a causare problemi, perciò è necessario che il programma che state eseguendo possa sfruttarne il più possibile. Periferiche quali i disk drive aggiuntivi richiedono memoria, ed andrebbero scollegate se non sono strettamente necessarie. Inoltre, bisogna rendere disponibili al programma tutte le librerie ed i driver dei quali ha bisogno; un software di comunicazione, ad esempio, richiede sicuramente che sul disco di sistema sia presente il file «devs/serial.device». Se non riuscisse ad avere accesso al file, è probabile che la temuta finestra del Guru faccia la sua comparsa.

Alcuni programmi, dopo essere stati eseguiti, non riportano il sistema alle condizioni in cui si trovava in precedenza: continuano cioè ad occupare memoria o ad inibire l'accesso ad alcuni files. Sarebbe bene quindi resettare il computer in alcuni ca-

INTERPRETAZIONE DEL NUMERO DI UNA GURU

```
Software Failure.          Press left mouse button to continue.
           Guru Meditation #07010001.00009310


  Specific error code -----+             +--- Task Address
                           |             |
                 07 01 0001 . 00009310
                  |  |
  Subsystem number -+  +--- General error code
```

FIG. 1

```
DOS library error = no memory at startup
```

```
SUBSYSTEM ERROR CODES
Exec Library          01      Console Device       11
Graphics Library      02      GamePort Device      12
Layers Library        03      Keyboard Device      13
Intuition Library     04      TrackDisk Device     14
Math Library          05      Timer Device         15
CList Library         06      CIA Resource         20
DOS Library           07      Disk Resource        21
RAM Library           08      Misc Resource        22
Icon Library          09      BootStrap            30
Audio Device          10      Workbench            31

GENERAL ERROR CODES

Non applicabile         00    OpenDevice error     04
Memoria insufficiente   01    OpenResource error   05
MakeLibrary error       02    I/O error            06
OpenLibrary error       03    No signal            07

SPECIFIC ERROR CODES

Exec Library
81000001    CPU trap vector checksum
81000002    ExecBase checksum error
81000003    library checksum failure
81000004    no memory to make library
81000005    corrupted free memory list
81000006    no memory for interrupt servers
81000007    InitAPtr
81000008    semaphore corrupted
81000009    free twice
8100000A    bogus exception

Graphics Library
82010001    copper display list, no memory
82010002    copper instruction list, no mem.
82000003    copper list too long
82000004    copper intermediate list too long
82010005    copper list head, no memory
82010006    long frame, no memory
82010007    short frame, no memory
82010008    flood fill, no memory
02010009    text, no memory for TmpRas
8201000A    BltBitMap, no memory
8201000B    region memory

Intuition Library
84000001    unknown gadget type
```

I CODICI DI ERRORE

FIG. 2

si, prima di procedere al caricamento di altro software, per evitare spiacevoli sorprese.

Ricordate infine che ci sono programmi sviluppati per funzionare su versioni specifiche dell'hardware di Amiga: software europeo, progettato per il funzionamento con macchine dotate di video secondo lo standard PAL, può creare problemi se fatto girare su computer americani, che seguono lo standard NTSC, e viceversa.

È sufficiente attenersi a queste semplici precauzioni per scongiurare, nella maggior parte dei casi, il pericolo di una visita indesiderata del Guru.

Per poter lavorare tranquillamente senza il timore che un blocco del sistema possa portare alla perdita dei dati o del programma presente nella macchina in quel momento, è possibile cautelarsi utilizzando il programma GOMF.

```
84010002    create port, no memory
84010003    item plane alloc, no memory
84010004    sub alloc, no memory
84010005    plane alloc, no memory
84000006    item box top < RelZero
84010007    open screen, no memory
84010008    OpenScreen's AllocRast, no mem.
84000009    open sys screen, unknown type
8401000A    add SW gadgets, no memory
8401000B    open window, no memory
8400000C    Bad State Return entering Int.
8400000D    Bad Message received by IDCMP
8400000E    Weird echo causing problem
8400000F    couldn't open the Console

DOS Library

07010001    no memory at startup
07000002    EndTask didn't
07000003    Qpkt failure
07000004    Unexpected packet received
07000005    Freevec failed
07000006    Disk block sequence error
07000007    Bitmap corrupt
07000008    Key already free
07000009    Invalid checksum
0700000A    Disk Error
0700000B    Key out of range
0700000C    Bad overlay

RAM library
08000001    Bad segment list

Expansion library
0A000001    Bad expansion free

TrackDisk Device
14000001    calibrate: seek error
14000002    delay: error on timer wait

Timer Device
15000001    bad request

Disk Resource
21000001    get unit: already has disk
21000002    interrupt: no active unit

BootStrap
30000001    boot code returned an error
```

SUL NOSTRO DISCO

Questo software di pubblico dominio, che trovate sul dischetto allegato a questo fascicolo di Amiga Byte, viene caricato al momento dell'accensione del computer e si installa in memoria, senza interferire con il funzionamento di altri programmi. Ogni qualvolta il sistema incappa in una Guru Meditation, GOMF prende il controllo e dà all'utente la possibilità di intervenire per evitare il blocco della macchina o consente di salvare il contenuto della memoria prima che vada perduto con il reset.

All'apparire della finestra di GOMF sarà possibile intervenire eliminando dalla memoria il programma colpevole di aver provocato il blocco del computer: selezionando l'opzione «Whap» e successivamente clickando nella finestra o nello schermo del programma in questione, GOMF elimina la causa del problema e riporta il sistema alle condizioni precedenti all'errore. Quindi, clickando sull'opzione «Gomf», si può riprendere il lavoro interrotto.

Può accadere che l'errore manifestatosi durante l'esecuzione del programma sia talmente grave da impedire anche questa funzione: in questo caso si riuscirà almeno a salvare il proprio lavoro su disco prima di procedere al reset.

I programmatori più esperti potranno includere parte del codice di GOMF all'interno del software da loro sviluppato: insieme al programma è infatti fornito un codice sorgente assembler, pronto per essere modificato ed utilizzato all'interno di altre applicazioni, in modo da renderle a prova di Guru.

CPU TRAPS

Bus Error	02
Address Error	03
Illegal Instruction	04
Divide by Zero	05
CHK instruction	06
TRAPV (Overflow)	07
Priviledge Violation	08
Instruction Trace	09
Opcode 1010	0A
Opcode 1111	0B
TRAP 0 ... 15	0A... 2F

FIG. 3

Il curioso nome GOMF è composto dalle iniziali dell'espressione gergale americana «Get Outa My Face!», che potremmo tradurre liberamente con «Fuori dai piedi!»; l'autore del programma infatti è stato spinto a realizzarlo perché esasperato dalle continue apparizioni di Guru Meditation che lo tormentavano mentre lavorava.

Lezioni di volo

di CARLO CATTONI

Impariamo a volare, quasi sul serio, con il Flight Simulator II. In questa prima lezione, l'uso degli strumenti ed il decollo.

Il Flight Simulator II della SubLogic è uno dei programmi più conosciuti dagli utilizzatori di personal computer; di fatto «il» simulatore di volo per eccellenza.

Le sue prestazioni sono il frutto di molti anni di sviluppo: la prima versione per Apple // e TRS 80 nacque nel '79, ma già nel '77 la SubLogic cominciava a sperimentare la sua caratteristica grafica vettoriale, studiata appositamente per le simulazioni. Ne sono state inoltre prodotte versioni adatte a tutti i modelli dei computer più diffusi; recentemente sono state introdotte quelle per la serie di macchine basate sul microprocessore MC68000, cioè il Macintosh, l'Atari ST e l'Amiga.

Recente è anche la commercializzazione dei dischetti scenario, che aggiungono altri aeroporti al database già ricco presente sul disco programma, e che permettono di coprire ormai per intero l'area degli Stati Uniti, mentre ne sono previsti di nuovi per altre zone.

La sofisticatezza raggiunta dal Flight Simulator II (FSII per gli amici) ha però un suo lato negativo: volare con il simulatore è diventato complicato quasi quanto volare sul serio; infatti i parametri presi in considerazione dal programma sono praticamente tutti quelli che contribuiscono a formare l'inviluppo di volo di un aereo leggero della classe dei 200 - 250 hp (in questo caso un Cessna 182 Turbo Skylane RG), e questo ha portato la FAA (la federazione aeronautica Americana) a consigliare caldamente alle scuole di volo l'utilizzo del FSII ai propri allievi come allenamento. Il realismo, quindi, è notevole; l'unica cosa della quale si sente veramente la mancanza è la sensazione fisica del volo: gli aerei, come e più delle auto, si pilotano anche con il ...sedere!

Quindi si è creata una situazione quasi paradossale: uno dei programmi più diffusi è anche uno dei meno sfruttati, dato il suo apprendimento obiettivamente difficile; sono stati scritti addirittura dei libri per completare le informazioni contenute nel manuale, ma a questo punto se l'interesse dell'utilizzatore si spinge tanto in là da voler conoscere proprio tutto sul volo, non resta che consigliargli un buon manuale di pilotaggio e suggerirgli di fare una visita al più vicino Aeroclub.

La situazione opposta è quella tipica della crisi di

rigetto dovuta a decine di miseri tentativi andati a vuoto: allora il mancato pilota abbandona qualsiasi velleità e bolla il simulatore come «gioco», ricordando tanto la favola della volpe e dell'uva... Se invece voi, ponendovi in una sana via di mezzo, desiderate apprendere quelle nozioni che vi permettano di decollare correttamente, volare verso una meta ed atterrare altrettanto correttamente (e non è per niente facile), allora queste semplici lezioni fanno al caso vostro.

DA DOVE COMINCIARE

Un punto di partenza obbligatorio imporrebbe la presenza del manuale, e la lettura della parte riguardante le possibilità di controllo del velivolo offerte dal programma utilizzando la tastiera, il joystick od il mouse; probabilmente la combinazione più efficace è quella che prevede l'uso del mouse per la manovra vera e propria e quello della tastiera per i comandi ausiliari.

Eviteremo lunghe discussioni teoriche sui termini aeronautici; questi verranno usati solo quando necessario ed eventualmente spiegati di volta in volta.

I nostri voli si svolgeranno per ora con tempo sereno, di giorno, e con i parametri di default impostati dal programma all'atto del caricamento: modo autocoordinato, moderato realismo, affidabilità totale e completa assenza di vento, nuvole o nebbia.

Eccoci quindi, caricato il programma, sulla testata della pista 27R dell'Oakland International Airport, a poca distanza da San Francisco; il motore è acceso, l'aereo è pronto per il volo; premete il tasto P per entrare in pausa.

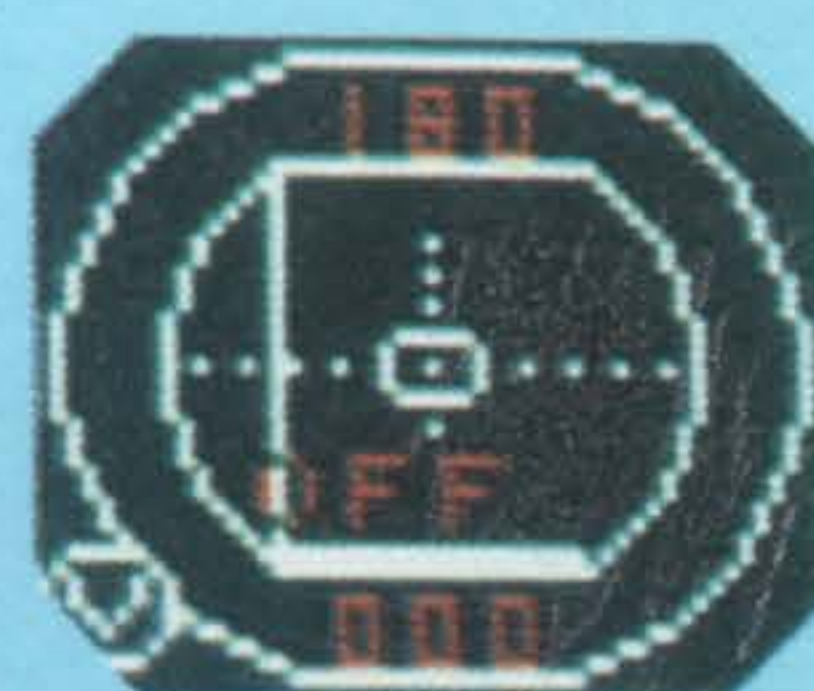
Facciamo subito un breve ripasso degli strumenti a nostra disposizione; gli otto strumenti principali rotondi sono, da sinistra a destra e dall'alto verso il basso: l'ANEMOMETRO,

che misura la velocità relativa all'aria esterna, NON al suolo, in Nodi (1 Nodo = 1 miglio nautico all'ora, cioè 1,85 km/h). Poiché voleremo in assenza di vento ed a quote non eccessive, potremo attenerci alle sue indicazioni senza bisogno di correzioni.

L'ORIZZONTE ARTIFICIALE: collegato ad un giroscopio che lo mantiene sempre in parallelismo costante rispetto al suolo; in pratica è come se l'aereo girasse intorno ad esso: ci fornisce indicazioni sul nostro assetto di volo. Tra questo e lo strumento successivo c'è l'indicatore di inserimento dell'autopilota.

L'ALTIMETRO: fornisce la nostra quota; anche le sue indicazioni sono sensibili alla temperatura ed alla pressione dell'aria esterna, quindi vanno corrette di tanto in tanto pigiando col puntatore sul pulsante rotondo con la lettera A. La lancetta lunga ci dà l'indicazione delle centinaia di piedi, quella corta delle migliaia e la freccettina fornisce l'indicazione delle decine di migliaia (la vedremo raramente).

Un'indicazione equivalente alle «tre e mezza» di un orologio corrisponde ad una quota di 3500 piedi (1 piede = circa 0,3 metri).

Il VOR 1: serve per mostrare la rotta da seguire se si intendono utilizzare i radioaiuti; per ora non ne parleremo.

Il COORDINATORE DI VIRATA: in realtà è composto da una sagomina dell'aereo e da un tubicino di vetro con un liquido nel quale è immersa una pallina; questa dovrebbe rimanere al centro quando si vira, per indicare che l'aereo non «sbanda».

Nel modo autocoordinato non dovremo preoccuparcene.

La sagomina invece, quando la punta di un'ala è allineata con una delle due tacche L o R, indica che stiamo effettuando una virata completa di due minuti, ossia di 180 gradi/minuto.

Vediamo ora la bussola.

La BUSSOLA GIROSCOPICA: indica la nostra prua, ossia in pratica dove stiamo andando; il numero in basso segnala la nostra rotta reciproca, ovvero da dove stiamo venendo.

Le sue indicazioni vanno corrette ogni tanto con il tasto D, per allinearle con quelle della bussola magnetica (COMPASS).

Il VARIOMETRO: indica il nostro rateo di salita o di discesa in centinaia di piedi al minuto.

Il VOR 2 (vedi VOR 1).

Sulla destra del quadro strumenti troviamo invece le Radio di comunicazione (COM) e di navigazione (NAV), il quadro motore con giri, pressione e temperatura olio, i televel del carburante; quindi gli indicatori di posizione di flaps e carrello, dei magneti motore, del riscaldamento del carburatore da usare in caso di formazione di ghiaccio, delle luci strumenti, l'orologio, il DME (Distance Measuring Equipment), l'ADF (Automatic Direction Finder), i marker dei sentieri di discesa (OMI) ed il Transponder. Gli indicatori dello zoom e della situazione del mouse ci sono necessari ma ovviamente non fanno parte della strumentazione di un velivolo.

IL TASTO GIUSTO

Generalmente ogni strumento contiene l'indicazione del tasto che serve ad utilizzare l'apparecchiatura monitorizzata (per esempio il carrello è esteso o retratto tramite il tasto U). È importante imparare ad utilizzarli: provateli!

Ricordate inoltre che premendo Help e clickando sullo strumento desiderato apparirà una finestra di aiuto.

Bi-clickando con il tasto sinistro del mouse (o clickando una sola volta con il destro) il puntatore scom-

pare: siamo ora in modo «yoke», ed il mouse viene usato come cloche o barra di comando: avanti per picchiare, indietro per cabrare, a sinistra ed a destra per virare. I suoi movimenti vengono riportati da cursori che si muovono su quella specie di grossa «I» situata in mezzo ai quattro strumenti della fila inferiore: la barra verticale indica la posizione dei timoni di profondità (o di quota, o anche equilibratori); quelle orizzontali superiore ed inferiore indicano la posizione degli alettoni e del timone di direzione; nel modo autocoordinato questi ultimi si muovono insieme, per permettere appunto il coordinamento nelle virate.

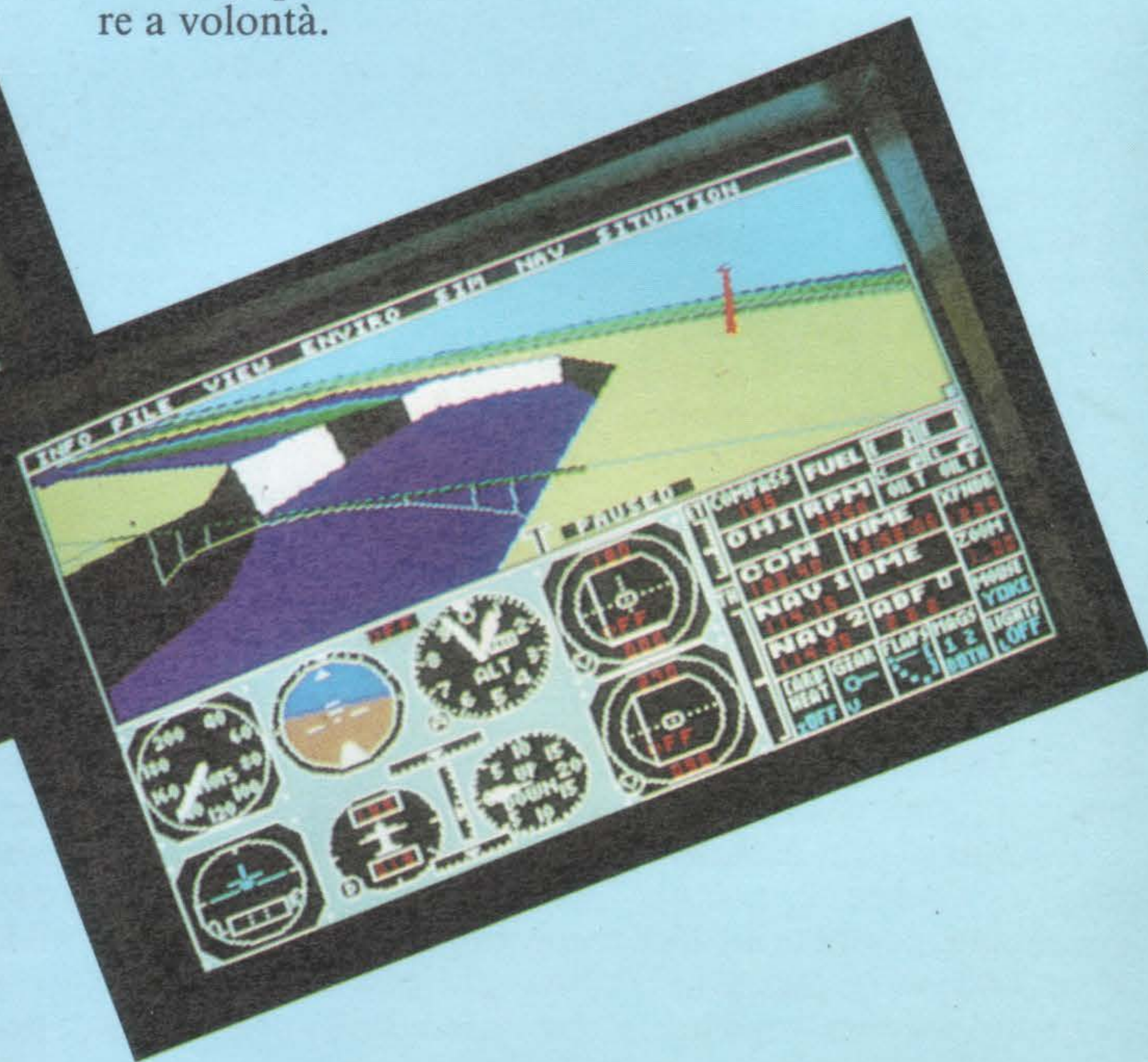
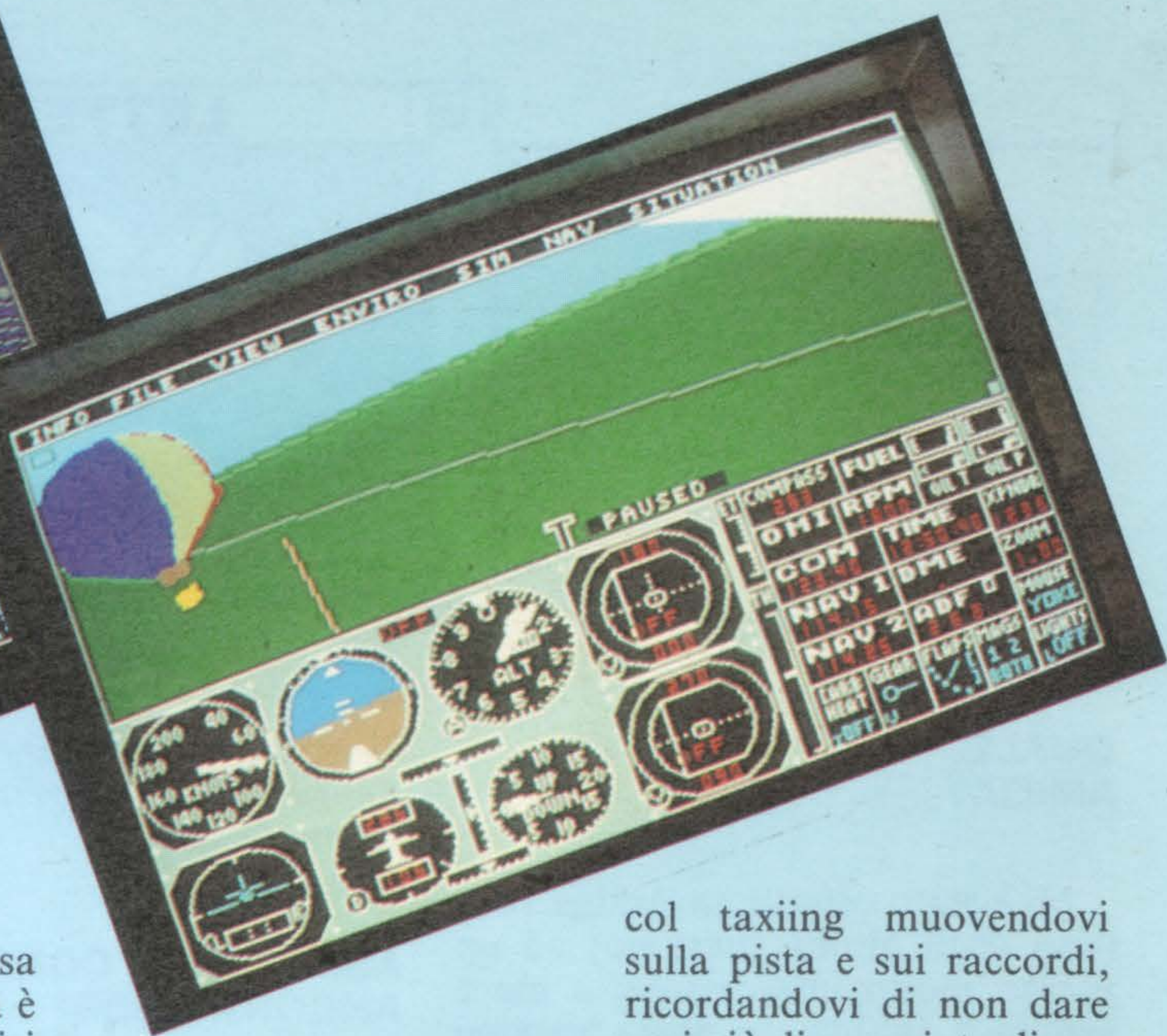
Spostate ora il mouse in su ed in giù tenendo premuto il pulsante sinistro: noterete come il cursore della manetta del gas (throttle) situato alla destra del VOR 2 si muove di conseguenza; fate pratica con la quantità degli spostamenti e le indicazioni fornite dal contagiri.

Tornate ora in modo cursore, clickando con il tasto destro o biclickando con il sinistro, e date un'occhiata ai vari menù per rendervi conto delle opzioni offerte. Ma NON VARIATE NULLA, almeno per ora. Noterete che di fianco a molte voci è presente l'indicazione del tasto che serve ad attivarle: imparate a farlo, perché durante il volo è importante cercare di distogliere il meno possibile il mouse dalle funzioni assegnategli di piccolo volante. Utilizzateli ora per variare la vostra vista: potete guardare intorno su 360 gradi a scatti di 45 con i tasti R, T, Y, F, H, V, B ed N, e guardare verso il basso con il tasto G.

Movimenti più fini, laterali o verticali, si ottengono con i tasti cursore. Fate qualche esperimento; quando avrete capito il meccanismo, premete T per tornare alla vista anteriore e Del per riportare il vostro punto di vista come all'inizio. Provate le viste esterne; lo Spot (tasto S) vi permette di osservare quello che fate come se foste su un altro aereo che vi segue, ed è molto importante. Tornate quindi nuovamente alla vista anteriore (tasto T).

Familiarizzate con le finestre: quella con la vista 3d secondaria e quella, più importante, contenente la mappa della zona in cui siamo; premete in questo caso F3 per ottenerla, ed F9 o F10 per ottenere lo zoom, cioè per ingrandire o rimpicciolire la zona osservata, come se si cambiasse quota. Potrete quindi osservare dall'alto la pista, l'aeroporto o la baia di S. Francisco; le finestre si comportano praticamente come quelle del Workbench, e si possono dimensionare, spostare e chiudere a volontà.

È importante osservare e magari tenere aperta la mappa dell'aeroporto quando si esegue il taxiing, ovvero i movimenti a terra necessari per portarsi in testata pista dal parcheggio o dalla zona rifornimento; questo perché la limitata risoluzione verticale del computer rende difficile percepire accuratamente la profondità e la distanza degli oggetti visti dalla stessa altezza. In volo le cose cambiano. È

giunta l'ora di muoversi: innanzitutto controllate che le vostre prove precedenti non abbiano lasciato il carrello in posizione retratta; se così fosse, premete U per estenderlo. Quindi verificate che il gas sia

completamente chiuso, le luci spente ed il riscaldamento del carburatore disinserito. Ora premete P per uscire dal modo pausa.

Un consiglio: usate spesso questo modo quando, durante il volo, ne sentirete la necessità; usate anche, dal menu «situation», il «select prerecorded/ Oakland 27R» per tornare, volendolo, alla situazione iniziale.

Osserviamo gli strumenti: tutti a zero. La bussola segna 276 gradi (la sigla 27R della pista indica un orientamento della stessa di 270 gradi, e la testata è quella destra); il contagiri 650 giri/min; i serbatoi sono pieni e l'olio è a posto; questo livello di realismo non contempla la necessità di usare i magneti per accendere e spegnere il motore.

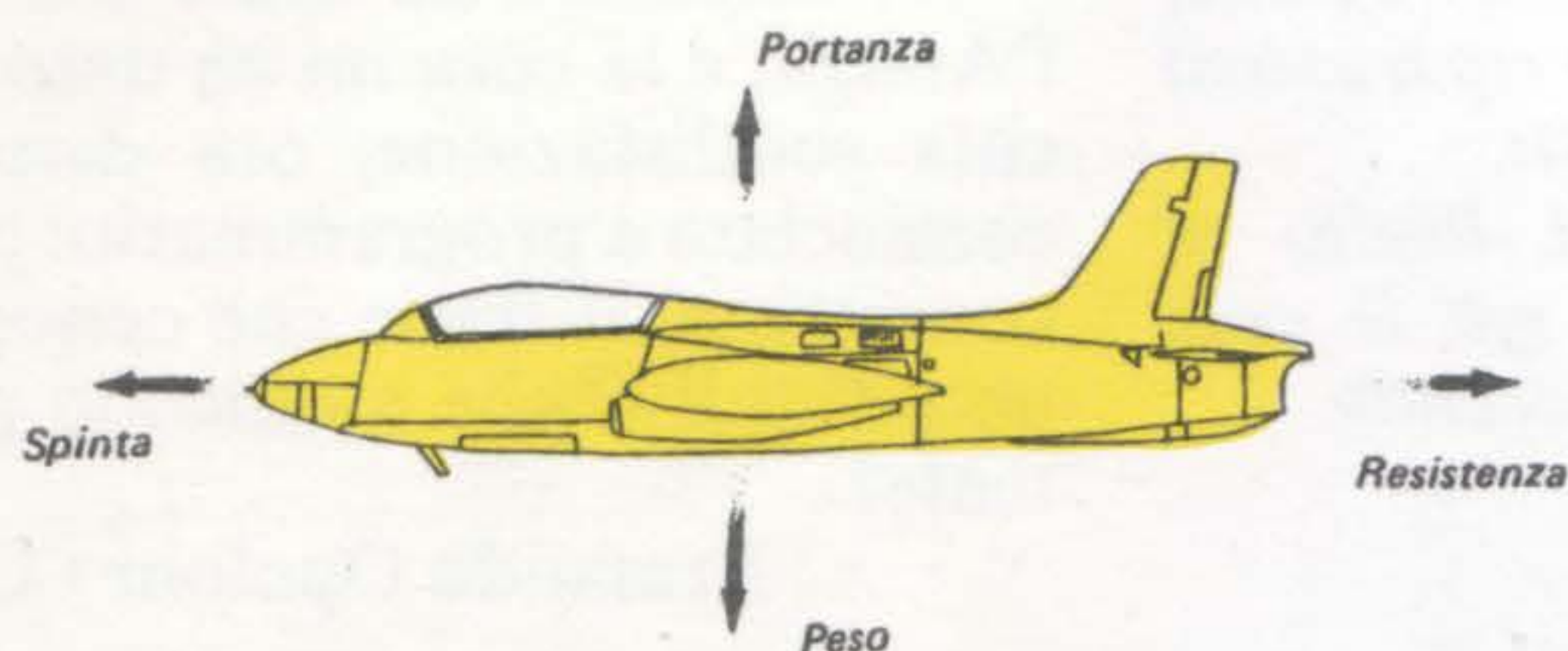
Mettete il mouse in modo «yoke» e spostatelo in modo che i cursori di alettoni ed il timone di direzione siano tutti a sinistra. Portate ora la manetta del gas fino ad un quinto circa della sua corsa: l'aereo comincerà a girare su se stesso.

Prima che la curva sia completa, togliete il gas e, quando la bussola segna ancora circa 270 gradi, premete il tasto 5 del tastierino numerico: questo è il modo più semplice e rapido per riportare al centro alettoni e timone di direzione. Premete quindi ripetutamente il tasto 1, che serve ad applicare i freni. Dovreste trovarvi adesso in una situazione simile a quella di partenza.

Fate un po' di pratica col taxiing muovendovi sulla pista e sui raccordi, ricordandovi di non dare mai più di un quinto di gas per non eccedere con la velocità (se necessario, ogni tanto toglietelo completamente) e di utilizzare la vista della mappa per tornare poi al punto iniziale. Se doveste perdervi, i modi per arrivarci sono due: selezionare «select prerecorded/Oakland 27R» dal menù Situation o premere U per far rientrare il carrello, affinché il velivolo si schianti sulla pista ed il programma faccia ripartire tutto da capo.

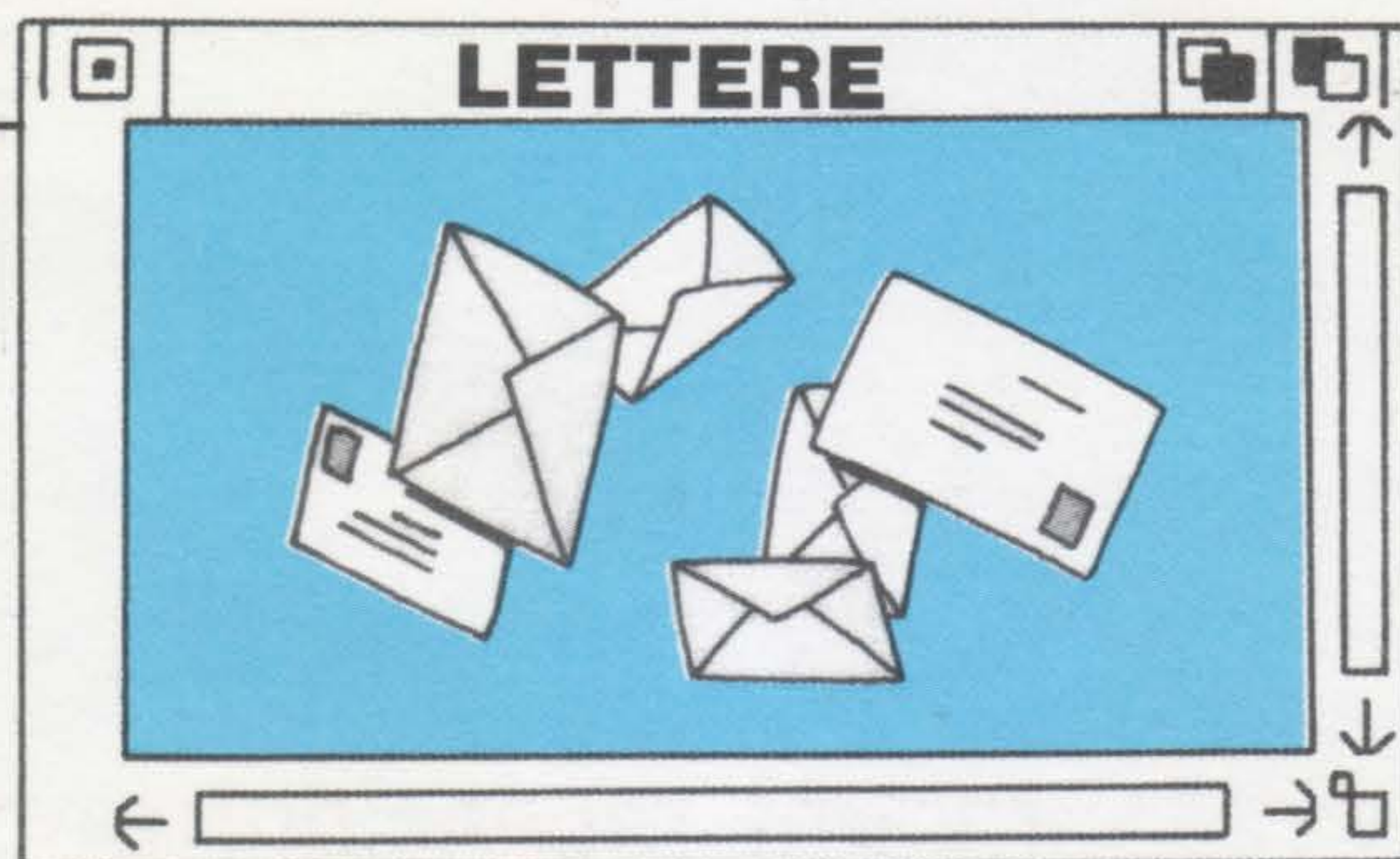
(continua)

CENNI DI FISICA DEL VOLO

La fisica del volo è molto complessa; si può dire che molti dei fenomeni riscontrati potranno in futuro essere spiegati più a fondo, data la continua evoluzione dei mezzi che la tecnica ci

Le forze che agiscono su di un aereo in volo rettilineo uniforme. Esistono altre forze, come le resistenze indotte, o quelle centrifughe che si verificano in caso di virata. Disegno da La Tecnica Aeronautica, Tripodi, Edizioni Delta.

mette a disposizione. Un brevissimo cenno può risultare comunque di aiuto a chi usa l'FSII.

Le forze che agiscono su di un aereo supposto in moto rettilineo uniforme sono quattro, e si annullano a due a due: spinta e resistenza, portanza e peso.

La portanza è creata dalle ali, specificamente dal moto relativo che esiste tra esse e l'aria: questa viene accelerata sul dorso dell'ala e rallentata sul ventre. Per il principio di conservazione dell'energia si crea così un differenziale di pressione a sfavore della parte superiore dell'ala, che in questo modo risucchia l'aereo verso l'alto, vincendo la forza peso. Un'altra componente della portanza è data dalla reazione dovuta alla deflessione dell'aria verso il basso; le superficie mobili dell'aereo servono a modificare la forma dell'ala per ottenere maggiore o minore portanza (i flaps), od accelerazioni del flusso aerodinamico per ottenere virate, salite o discese (alettoni, timoni di profondità o equilibratori, timone di direzione). Una maggiore portanza è necessaria in fase di atterraggio o di volo lento, per impedire lo stallo, che sopravviene principalmente quando il flusso aerodinamico intorno all'ala diviene scomposto causa la scarsa velocità, portando così ad una mancanza di sostentamento ed alla caduta. Lo stallo può verificarsi anche in velocità, quando l'angolo formato dall'ala e dal flusso aerodinamico relativo (angolo di attacco) assume valori troppo elevati; tutte situazioni da evitare assolutamente.

QUALE AMIGA?

Mi è stato offerto un Amiga 1000 (uno degli ultimi, presumo) ad un prezzo molto interessante; a me è venuto il dubbio che possa trattarsi di una fregatura, perché so che questo computer costava in precedenza molto di più, e che è stato soppiantato da nuovi modelli.

Vale la pena di acquistarlo o devo lasciar perdere?

Mario Citresa - Introbio (CO)

Sicuramente non si tratta di una fregatura, se il prezzo è veramente interessante! Tieni presente che il modello più economico, il 500, fa le stesse cose ma è ergonomicamente più sacrificato e l'impostazione generale è più spartana. Il vantaggio dei nuovi modelli è dato dal fatto che le prossime estensioni hardware saranno dedicate interamente a loro, e già ora è più facile espanderli.

Comunque, poiché esiste una base di Amiga 1000 molto ampia, il mercato per questa macchina è già ben sviluppato e tuttora in fase di crescita.

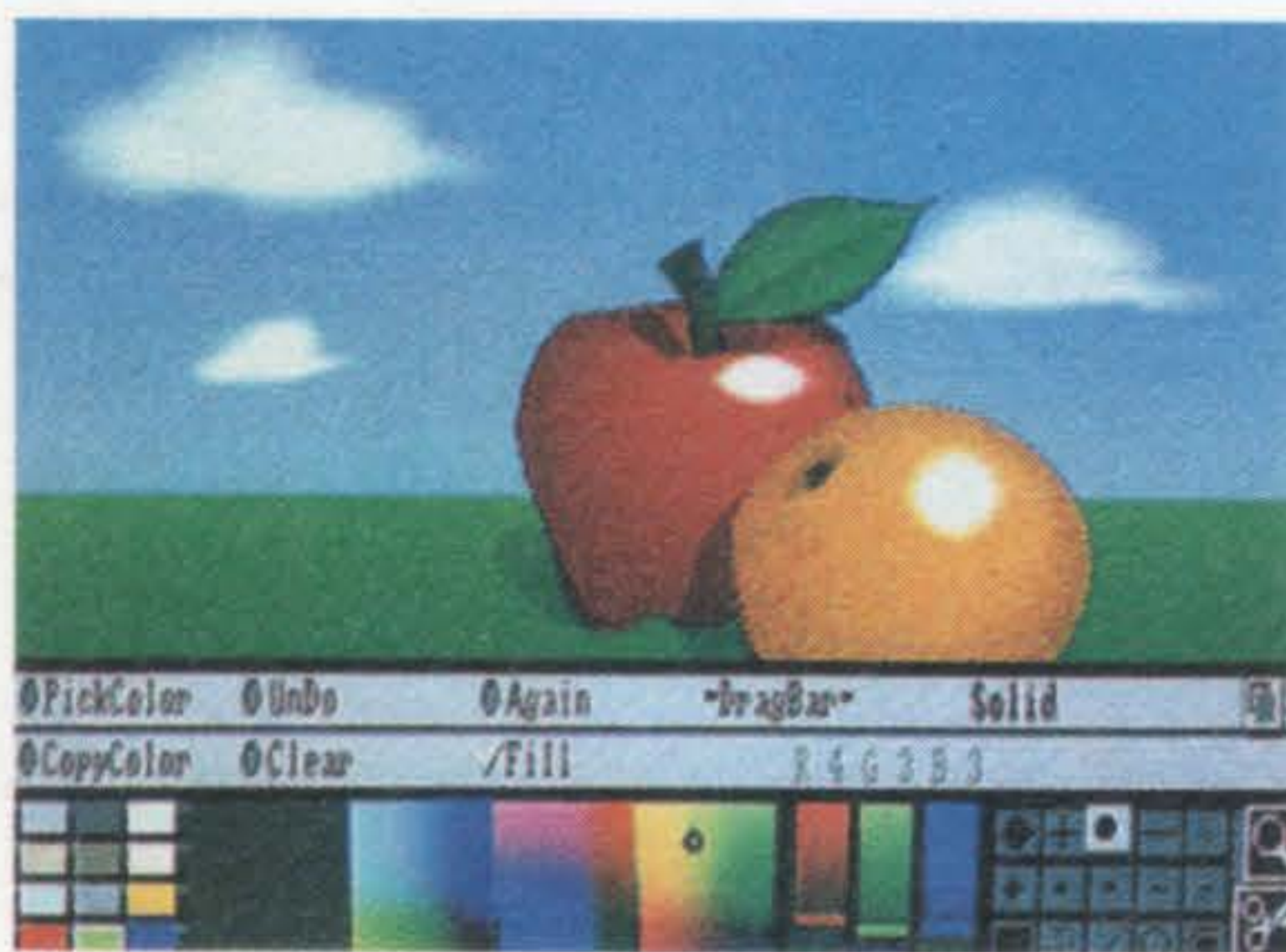
DRIVES ESTERNI

Possiedo un drive esterno da 3 pollici e mezzo autoalimentato che utilizzavo con il mio computer precedente, un QL; l'ho tenuto perché pensavo di poterlo collegare con il mio nuovo Amiga 500, ma ora non mi fido a collegarlo in quanto non so se è compatibile. Cosa posso fare?

Nino Ruotolo - Casalnuovo (NA)

I drives esterni per Amiga contengono un dispositivo per la selezione automatica del numero del drive; per collegare un drive non dedicato ti conviene attendere la prevista uscita su Amiga Byte del progetto di un'apposita interfaccia.

ASSISTENZA COMMODORE: ANCORA DUBBI

Da un po' di tempo sono tentato di comprare un Amiga, ma mi preoccupano le voci riguardanti l'assistenza fornita dalla Commodore, che non sarebbe in grado di riparare

i computer in tempi brevi. Dato che intenderei fare un uso professionale del mio personal, la cosa non può passare inosservata: cosa ne dite?

Giorgio Marconi - Codogno (MI)

La Commodore si sta muovendo con decisione in questo senso, dato lo sviluppo del mercato professionale. Ormai esiste almeno un centro di assistenza autorizzato per ogni regione, ed i tempi di attesa per le riparazioni si sono notevolmente ridotti.

Non siamo ancora al livello di Germania o Inghilterra, ma le cose stanno senza dubbio migliorando.

CHE STAMPANTE SCEGLIERE

Ricevuto in regalo un Amiga 2000, vorrei completare il sistema con una stampante di buona qualità: il mio problema riguarda la scelta, che è molto vasta e non so come orientarmi.

Ho visitato diversi negozi, ma la cosa non ha fatto altro che aumentare la mia confusione: prezzi diversissimi per lo stesso hardware, offerte speciali, un mare di marche tutte famose.

Dato che la mia esperienza informatica è scarsa, non vorrei fare una scelta sbagliata: potete consigliarmi?

Bartolomeo Nucci - La Spezia

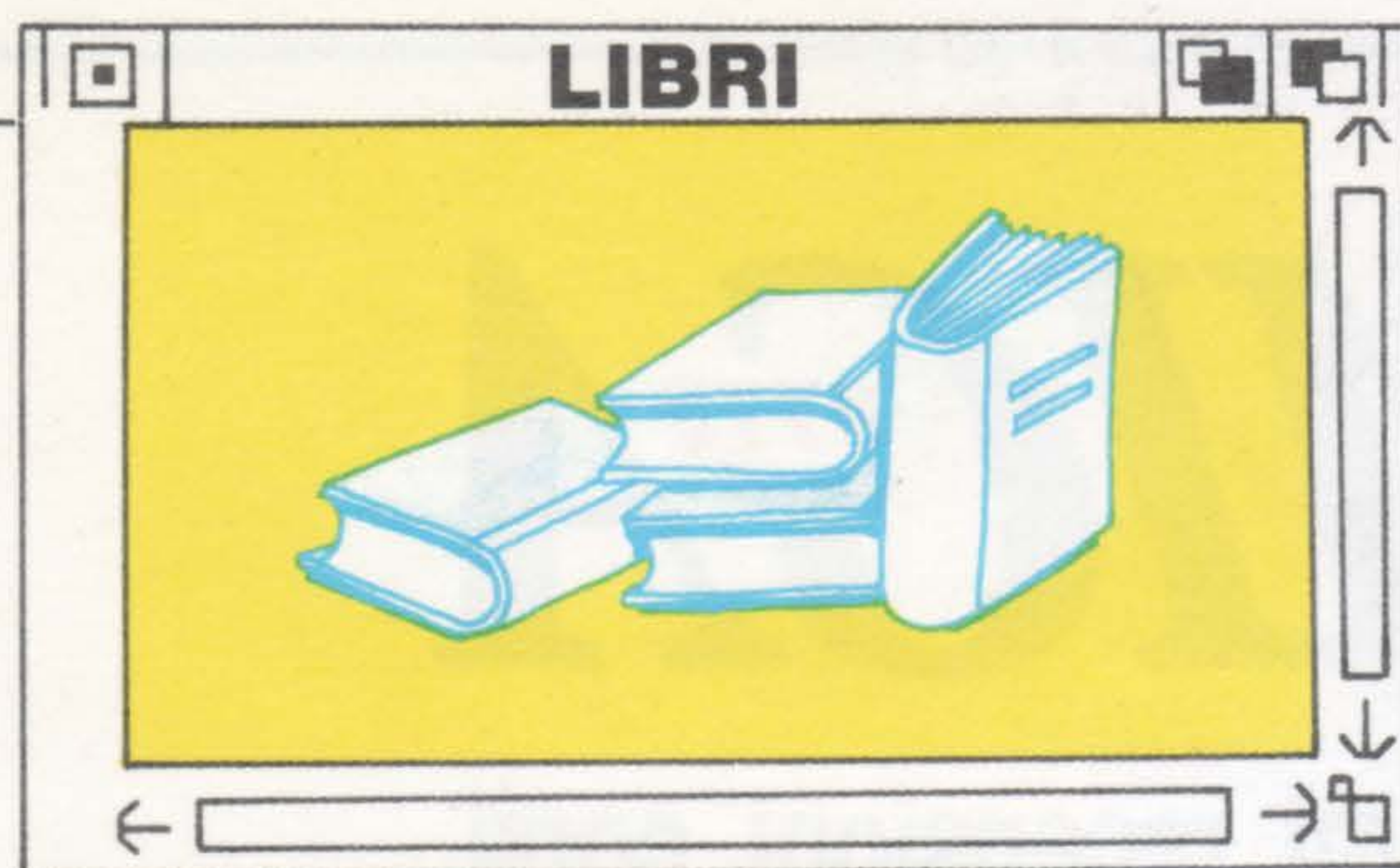
Prima di tutto dovresti avere ben chiare le tue esigenze: se desideri farne un uso professionale, la stampante dovrà essere adeguata alle tue necessità ed il prezzo sarà di conseguenza piuttosto elevato: affidati ad un'organizzazione seria che ti garantisca una buona assistenza, anche se il costo dovesse risultare più alto che altrove.

Se invece il tuo fine è hobbistico, allora puoi forse permetterti di essere meno esigente. Ricordati però di rivolgerti ad una marca che permetta la compatibilità con i drivers per stampante disponibili per l'Amiga (l'emulazione più comune è quella Epson).

BASIC OPPURE «C»?

Ho imparato ad usare il Dos dell'Amiga, e la cosa mi ha dato parecchia soddisfazione; ora desidererei cominciare a programmarlo: cosa mi consigliate, il Basic che conosco già un po' o il C che mi dicono più potente?

Armando Opoloni - Catania

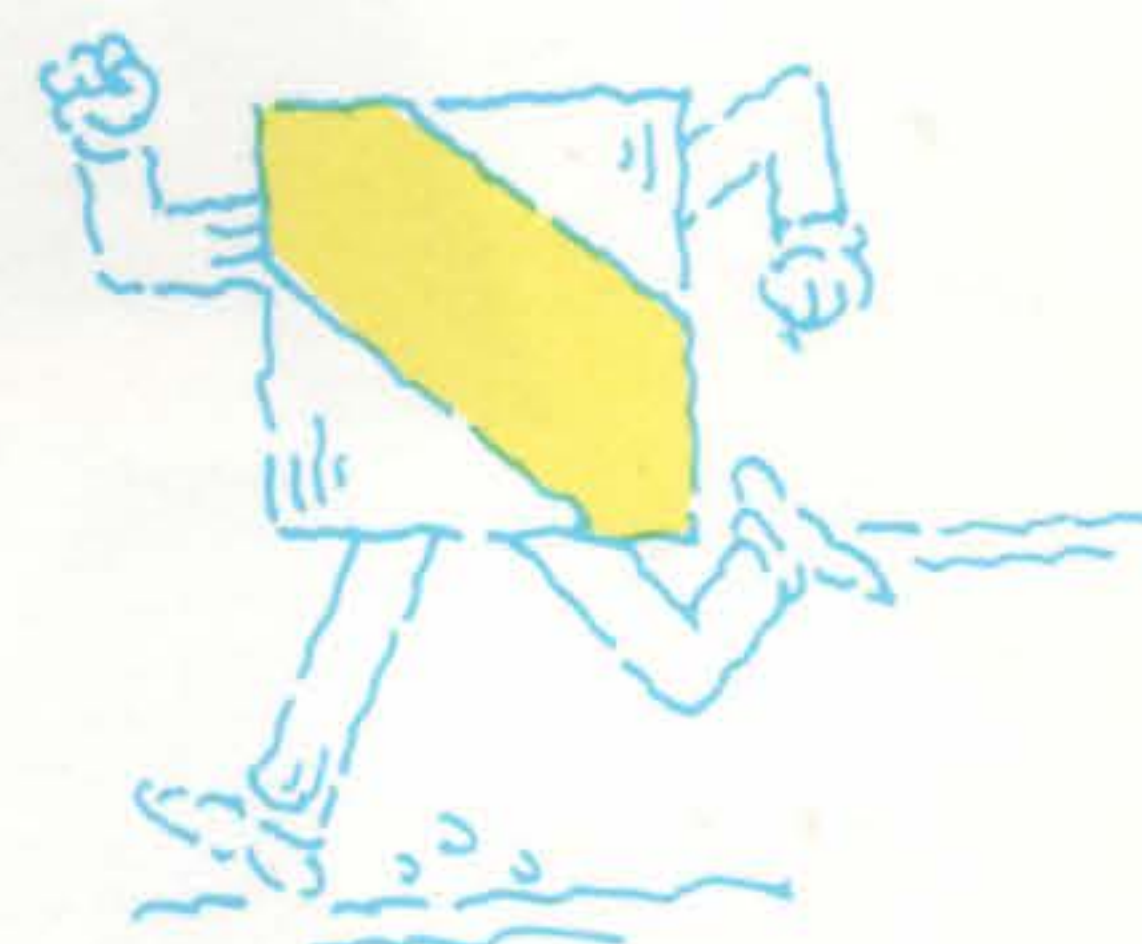
Decisamente il C è più versatile del Basic, ed è il linguaggio «di elezione» dell'Amiga. È anche, però, il più difficile da apprendere, ma ha l'indiscusso vantaggio di essere trasportabile da un computer ad un altro grazie all'intenso uso delle fantomatiche librerie di sistema.

AMIGA 500 GUIDA PER L'UTENTE

di R. Bonelli - M. Lunelli
358 pag. Lit. 55.000

Il titolo del libro rivela chiaramente l'intenzione degli autori di fornire agli utenti del nuovo Amiga una serie di informazioni che attualmente è possibile trovare soltanto in forma dispersa su diverse pubblicazioni. Certamente questa guida non si rivolge agli smanettoni, che sono la minoranza, ma alla maggior parte degli utenti bisognosa di un manuale comprensibile, soprattutto in italiano. Fondamentalmente il tutto si articola in due parti: la prima, di 5 capitoli, si occupa del sistema operativo mentre la seconda, di 3 capitoli, della programmazione in Basic.
Il primo ad essere esaminato è il mouse ed il suo utilizzo, soprattutto nei menu del sistema operativo Workbench, per poi passare ad analizzare i programmi di servizio contenuti nel dischetto di sistema (Preferences, Notepad, Nofastmem, etc.) e quelli presenti nella directory Tools del disco Extras (More, Printfiles, etc.). Non manca la sezione dedicata al CLI ed alla spiegazione (sufficientemente dettagliata) dei vari comandi del Dos con la relativa sintassi.
Successivamente viene fatta una panoramica dell'interprete Amiga Basic, con il relativo compilatore AC-Basic, e sul potente compilatore True Basic, dotato di un proprio editor. Attraverso vari esempi (oltre una trentina) vengono spiegate le istruzioni più complesse toccando tutte le caratteristiche fondamentali di Amiga quali il suono, la grafica e la gestione dei file. L'inserimento sistematico di semplici listati facilita moltissimo la comprensione dei concetti e l'unico inconveniente (per altro necessario!) sarà dato dalla battitura delle righe Basic.
Ottima l'idea di trattare anche dei due compilatori, cominciando da quello strettamente legato all'Amiga Basic e riportando le differenze esistenti tra i vari linguaggi sia a livello di benchmark (cioè rapportando le velocità di esecuzione) sia a livello formale (cioè di formato delle istruzioni). Due schede riassuntive descrivono in ordine alfabetico tutti i comandi del Basic e del True Basic alla fine dei rispettivi capitoli, visto che non fà mai male avere sotto gli occhi tutte le sintassi. È un ottimo testo che si propone quasi come manuale dell'Amiga 500, dedicato a tutti gli Amighi neofiti e non, che costituisce senza dubbio un primo mattone per la scalata al vertice della potenza di Amiga. Unico neo, il prezzo un po' elevato.

ROM KERNEL REFERENCE MANUAL: LIBRARIES AND DEVICES

Addison Wesley editore
Lit. 72.000

Questo libro fà parte di una collana di quattro tomi tutti editi da Addison Wesley, collana considerata la «bibbia» del programmatore di Amiga. Confermiamo questo giudizio anche noi e, a scanso di equivoci, avvertiamo il lettore che questo volume (circa 1100 pagine!) di routine e disassemblati è riservato soltanto a chi vuole approfondire la programmazione. In realtà chiunque avesse intenzione di programmare seriamente non potrebbe farlo se non conoscesse le varie librerie, il loro significato ed il loro uso. Non pensate però che sia un discorso riservato esclusivamente agli eroi dell'Assembler o del C, poiché è possibile usare le librerie anche dal Basic. Ovviamente il libro è tutto in inglese ed è stato scritto dagli sviluppatori hardware-software dell'Amiga stesso già, questa, garanzia di serietà. Sfogliandolo troviamo una introduzione generale all'uso delle librerie, centinaia di esempi per illustrare la funzione delle routine presenti in Rom (192 K di codice), un tutorial su grafica ed animazione ed una lista completa dei devices e delle librerie della Rom. Nonostante sia doveroso denunciare, in alcuni punti, un discorso un po' intricato e di difficile comprensione, è senza dubbio un testo fondamentale, diremmo indispensabile per i programmatori. Vengono svelati molti misteri attraverso semplici esempi, tutti rigorosamente in C; ovviamente è sottinteso che il lettore abbia una infarinatura piuttosto approfondita di questo linguaggio, che è usato tranquillamente come se fosse italiano. Purtroppo tutte le informazioni sono riferite alla versione 1.1 del Kickstart ed ancora non è ufficiale l'uscita della versione aggiornata, ma per la comprensione della logica e del funzionamento delle varie librerie e devices, è più che sufficiente.

Per una completa conoscenza della macchina sarebbe necessario leggere tutti i libri della collana, visto che toccano argomenti fondamentali di Amiga quali l'hardware, Intuition e l'Exec, ma questo Rom Kernel è senza dubbio il primo da interpellare per poter capire meglio il nostro computer. ■

ASSEMBLER

SPOSTIAMO I DATI

2ª lezione

di LEONARDO FEI

Iniziamo, questa volta, ad esaminare il set di istruzioni e, parallelamente, i modi di indirizzamento del 68000. Questo microprocessore è caratterizzato da un set di istruzioni base, circa una cinquantina, sulle quali si possono applicare un notevole numero di indirizzamenti diversi, creando in definitiva un numero di istruzioni utilizzabili molto ampio. Cominciamo dalla più classica ed usata istruzione, quella di spostamento: MOVE. La sintassi è la seguente:
MOVE.e <arg1>,<arg2>
dove la e può essere una delle seguenti lettere di estensione: B, W, L, e dove <arg1> e <arg2> sono i due modi di indirizzamento che si possono scegliere nella seguente tabella:

<arg1>:	
x Dn	**x s8(An,Xn)**
x An	**x xxxx.W**
x (An)	**x xxxx.L**
x (An)+	**x s16(PC)**
x —(An)	**x s8(PC,Xn)**
x s16(An)	**x #<dati>**
<arg2>:	
x Dn	**x s8(An,Xn)**
An	**x xxxx.W**
(An)	**x xxxx.L**
x (An)+	**s16(PC)**
x —(An)	**s8(PC,Xn)**
x s16(An)	**#<dati>**

Il significato dei simboli che compaiono nella tabella è spiegato in figura A. Se non compare una x accanto al simbolo, vuol dire che quell'indirizzamento non può essere utilizzato.

Cominciamo con l'indirizzamento Registro Data diretto:

MOVE.L D0,D1
copia il contenuto del registro D0 nel registro D1. Tutti i 32 bit vengono copiati, perché abbiamo specificato l'estensione .L, che significa longword (32 bit, appunto).

Avremmo potuto specificare anche: .B (byte, 8 bit) e .W (word, 16 bit). In mancanza della lettera di estensione, i comandi si comportano come se fosse stata specificata la lettera W, operando su word.

Registro Address diretto:
MOVE.L A3,D0
copia il contenuto del registro A3 nel registro D0. Notiamo che non può essere effettuata l'operazione inversa, cioè MOVE.L D0,A3, in quanto nella tabella di indirizzamento di MOVE, nella parte relativa ad <arg2>, non è presente la x accanto ad An (Registro Address diretto). Per spostare qualcosa in un registro Address, bisogna utilizzare l'istruzione MOVEA, di cui parleremo in seguito.

Registro Address indiretto:
MOVE.L (A3),D4
copia il contenuto della locazione di memoria puntata da A3 (il cui indirizzo è contenuto in A3) in D4. In pratica questa istruzione legge il contenuto di A3, lo interpreta come indirizzo assoluto, e copia il contenuto della memoria da questo indirizzo in D4.

Registro Address indiretto con postincremento:
MOVE.L D5,(A5)+
copia il contenuto di D5 nella locazione di memoria puntata da A5, poi aggiunge 4 ad A5, in modo che punti alla longword successiva in memoria. Se avessimo specificato l'estensione .B, il contenuto del registro A5 sarebbe stato incrementato di 1, per puntare al byte successivo, mentre con .W sarebbe stato incrementato di 2.

Registro Address indiretto con predecremento:
MOVE D6,—(A2)
prende il valore contenuto in A2, lo decrementa di 2 (nessuna estensione = .W) e memorizza, all'indirizzo così ottenuto, la word contenuta nella metà inferiore di D6.

Registro Address indiretto con spiazzamento:
MOVE.B D0,34(A2)
prende il contenuto di A2, vi somma 34, e memorizza, all'indirizzo così ottenuto, il primo byte di D0.

Registro Address indiretto con indice:
MOVE.L D0,20(A2,D3.W)
prende il contenuto di A2, vi somma lo spiazzamento 20, quindi vi somma la word inferiore di D3 (l'indice), e memorizza all'indirizzo ottenuto il registro D0.

Assoluto Corto:
MOVE.W D5,$700
copia la word di D5 all'indirizzo assoluto $700.

Assoluto Lungo:
MOVE.B D4,$70000
copia il byte di D4 all'indirizzo assoluto $70000.

Program Counter con spiazzamento:
MOVE.L 34(PC),D6
copia la longword, puntata dal Program Counter addizionato di 34, nel registro D6.

Program Counter con indice:
MOVE.W 24(PC,D3.L),D5
copia la word, puntata dal PC al quale vengono sommati lo spiazzamento ed il contenuto dell'indice D3, nel registro D5.

Immediato:
MOVE.W #$FE80,D2
mette il valore assoluto $FE80 in D2.

Fermiamoci un attimo per esaminare meglio le specifiche dei modi di indirizzamento. Quando utilizziamo lo spiazzamento, rispetto ai registri address o rispetto al PC, abbiamo 16 bit a disposizione. Questo significa che il raggio d'azione è limitato fra —32768 e +32767 bytes, rispetto ad An o PC.

Se utilizziamo l'indirizzamento con indice, lo spiazzamento è limitato ad 8 bit (fra —128 e +127), mentre del registro indice possiamo utilizzare, la word o la longword. Per capire meglio come funziona tutto questo, è essenziale cominciare a vedere qualche esempio. Vediamo quindi quali programmi possiamo utilizzare per assemblare i nostri sorgenti. Chi possiede il compilatore C della Lattice o della Manx, può utilizzare l'assembler che entrambi questi prodotti offrono. L'assembler della Lattice si chiama ASM, quello della Manx si chiama AS, ed entrambi si trovano nella directory che contiene il compilatore C. Le differenze fra questi due assemblatori ed il loro utilizzo non sono rilevanti, e vengono ben documentate nei rispettivi manuali. Una volta assemblato il codice sorgente nel file oggetto, bisogna utilizzare i rispettivi linker (Blink per il Lattice, e LN per il Manx) per ottenere l'eseguibile. Ma per i nostri programmi iniziali, che non conterranno subito chiamate a librerie di sistema, una soluzione sicuramente più comoda è rappresentata dal Seka Assembler. Questo assemblatore può generare sia il file oggetto, da linkare successivamente con l'Alink o il Blink, sia direttamente l'eseguibile. La comodità offerta è quella di poter provare i nostri programmi subito dopo l'assemblamento, senza perdere tempo nella fase successiva. Partiamo quindi dal presupposto di utilizzare questo assembler (benché nulla vieti l'utilizzo di un qualsiasi altro analogo prodotto). Lavoreremo sempre nella CLI, quindi potete evitare di caricare il Workbench, per risparmiare memoria e tempo prezioso. Il nostro primo approccio con l'assembly non può consistere nel classico programma che stampa sulla schermo una frase, perché per effettuare questa operazione è necessario un lavoro relativamente complesso, rispetto a quello richiesto dal Basic o dal C (PRINT e Printf). L'unica operazione semplice che possiamo effettuare, e di cui possiamo avere un riscontro nel «mondo esterno», consiste nello scrivere un certo valore in una locazione di memoria che sia collegata a qualcosa osservabile. Quale scelta migliore del led di alimentazione? Il sorgente del programma si trova sul disco allegato a questo fascicolo di Amiga Byte nella directory corso_asm, sotto il nome «prog. 1»:

```
start:
MOVE.B #$02,$BFE001        ; setta il bit 1 (/LED=1)
MOVE.W #$FFFF,D0
ritardo:
DBRA     D0,ritardo        ; ciclo di ritardo
MOVE.B #$00,$BFE001        ; pulisce il bit 1 (/LED=0)
RTS                        ; fine
```

Scriviamo il programma con un text editor e carichiamo il Seka Assembler. Alla richiesta «WORKSPACE KB>», rispondiamo con il numero di Kilobytes da dedicare alla memoria di lavoro (10 o un altro numero a nostra scelta, a seconda della disponibilità di memoria). Carichiamo il sorgente con il comando R (read), ed assembliamolo con il comando A (assemble), battendo RETURN alla richiesta «OPTIONS>». Se ci sono errori di battitura, possiamo correggerli utilizzando l'editor interno del Seka (premendo Esc), oppure possiamo cancellare il sorgente in memoria con il comando KS (kill source), rieditarlo e salvarlo con il text editor e ricominciare da capo. Una volta assemblato correttamente (messaggio «No Errors»), salviamo l'eseguibile con il comando WO (write object), e proviamolo. Subito dopo aver premuto il tasto RETURN per lanciare il programma, fissiamo lo sguardo sul LED di alimentazione di Amiga. Per un brevissimo intervallo si abbasserà la luminosità. Ma cosa è successo? All'indirizzo assoluto $BFE001, c'è una porta ad 8 bit che, fra le altre cose, controlla la luminosità del LED di alimentazione. È proprio questa porta che viene utilizzata durante un GURU od un RESET per fare lampeggiare il LED in questione. Il bit che lo controlla è il numero 1 (secondo bit a partire da destra), al quale corrisponde il valore decimale ed esadecimale 2. Normalmente questo bit è basso (0), ed il LED è luminoso. Se alziamo questo bit (1), il LED diventa scuro, e questa è l'operazione svolta dalla prima istruzione: MOVE.B #$2,$BFE001., quella di mettere il valore immediato $2 nella locazione assoluta $BFE001. La seconda istruzione carica il valore immediato $FFFF (decimale 65535) nel registro D0. Questo registro è utilizzato come contatore per il ciclo di ritardo realizzato dalla terza istruzione DBRA (che in alcuni assembler viene chiamata DBF). Questa istruzione testa il registro D0 per vedere se il suo contenuto corrisponde alla condizione specificata (DBF=Decrement and Branch False: decrementa e salta fino a quando il valore di D0 è 0). D0 contiene 65535, quindi il suo valore è decrementato di uno, ed il programma salta alla label (etichetta) «ritardo». Questo si ripete fino a quando D0 non ha raggiunto il valore 0, quindi per 65536 volte, poi il programma prosegue all'istruzione successiva, MOVE.B #$00,$BF001, che rimette zero nel bit 1, accendendo il Led. Vediamo ora il seguente programma:

Registro Data diretto	**Dn**
Registro Address diretto	**An**
Registro Address indiretto	**(An)**
Registro Address indiretto con postincremento	**(An)+**
Registro Address indiretto con predecremento	**—(An)**
Registro Address indiretto con spiazzamento	**s16(An)**
Registro Address indiretto con indice	**s8(An,Xn)**
Assoluto Corto	**xxxx.W**
Assoluto Lungo	**xxxx.L**
Program Counter con spiazzamento	**s16(PC)**
Program Counter con indice	**s8(PC,Xn)**
Immediato	**#<dati>**

Figura A. Modi di indirizzamento del 68000 e relativi simboli.

```
start:
MOVEQ     #$0F,D1           ;contatore del ciclo esterno
ciclo:
MOVE.B    #$02,$BFE001      ;setta il bit 1 (/LED=1)
MOVE.W    #$FFFF,D0
ritardo:
DBRA      DO,ritardo        ;ciclo di ritardo
MOVE.B    #$00,$BFE001      ;pulisce il bit 1 (/LED=0)
ritardo2:
DBRA      D0,ritardo2       ;ciclo di ritardo2
DBRA      D1,ciclo          ;fine del ciclo esterno
RTS                         ;fine
```

La prima istruzione è una variante dell'istruzione MOVE.
La sintassi è la seguente:
MOVEQ #<dati>,Dn
dove #<dati> è un valore immediato ad 8 bit, che viene esteso di segno a tutti i 32 bit e memorizzato nell'intero registro Data. In pratica se il bit 7 (l'ottavo), che rappresenta il segno, è settato (1), vengono alzati tutti i 24 bit rimanenti, altrimenti sono settati a zero. Nel nostro caso il valore $08 viene messo nel byte inferiore del registro D1, ed i tre byte rimanenti sono messi a 0. Il LED viene «oscurato» dalla seconda istruzione, ed il programma esegue il primo ciclo di ritardo. Al termine di questo ciclo il led viene «illuminato» ed il programma esegue un secondo ciclo di ritardo. Viene quindi controllato se il ciclo esterno è finito, ed in caso contrario il programma salta a «ciclo» e tutta l'operazione si ripete. Come risultato finale, noi vediamo il LED lampeggiare 16 volte. ■

Amiga Team From Reggio Calabria cerca utenti, per scambio di ogni genere di programmi, di idee, opinioni, e tutto ciò che possa riguardare questo meraviglioso, stupendo e ineguagliabile computer. Non aspettare. Invia subito la tua lista a: Emanuele Marino, via Cardinale Portanova 53/A, 87100 Reggio Calabria (RC). Oppure telefona al 0965/98220.

Scambio software. Ultime novità chiedere lista. Massima serietà. Scrivere o telefonare a: Andrea Borri, via Cabagni Baccini 51, 18039 Ventimiglia (IM), tel. 0184/294761.

Informazioni e programmi cambio con utenti. Scrivere o telefonare a: Emiliano Bagnoli, via Mascagni 39, 41100 Modena (MO), tel. 059/237675.

Cerco software Amiga, Ibm ps/20 su dischetti da 3.5; se avete programmi di grafica ed ingegneria civile, sarei interessato a comprarli. Inviate le vostre liste a: Luigi Manfredi, via I Maggio 25, 88074 Crotone (CZ).

Attenzione! Cambio software di qualunque genere. Dispongo di oltre 300 programmi, con le ultime novità in commercio; massima serietà e risposta sicura. Telefonate o scrivete a: Claudio Sacco, via San Carlo 8, 28048 Verbania Pallanza (NO), tel. 0323/503132.

Manuali e programmi scambio. Telefonate o scrivete a: Federico Spinola, via Saffi 1, 40131 Bologna (BO), tel. 051/551213.

Club Verbania ha la disponibilità di centinaia di programmi e molti manuali. Se volete contattarci telefonate o scrivete a: Di Niozzi, via per Possaccio 66, 28059 Verbania (NO).

Per Amiga cerco e scambio manuali. Telefonatemi subito, aspetto. Carlo Mainardi, via Bologna 13, 47036 Riccione (RI), tel. 0541/42878.

Per scambio software cerco utenti, inviate le vostre liste a: Luigi Fortino, via Torino 3, 84092 Bellizzi (SA).

Avete un'Amiga 2000? Cerco, possibilmente in zona Pavia, amici per scambio programmi e informazioni. Il mio indirizzo e numero di telefono sono: Davide Manzella, via Sesia 3, 27029 Vigevano (PV).

Tastiera italiana Amiga 1000, in perfette condizioni, con moltissimo software selezionato e relativi manuali, vendo al miglior offerente. Compro «Blink» della Software Distillery versione 7.1 e manuali aggiornati del compilatore manx ultima varsione. Luigi Callegari, via De Gasperi 47, 21040 Sumirago (VA), tel. 0331/909183 (sera).

Sidecar compro, anche usato, max L. 500.000. Scambio software ed informazioni hardware Amiga, Ms-Dos, Apple. Valerio Bonetti, via Tomitano 13, 31100 Treviso (TV), tel. 0422/22350 (ore pasti).

Modem qualsiasi marca, in perfetto stato, compro. Solo prezzo interessante. Scrivere o telefonare a: Walter Gaspari, via Cesare Battisti 43, 24060 Chiuduno (BG), tel. 035/839151.

Possiedo tutti i dischi di Public Domain compresa la raccolta Fish completa e tutte le ultimissime e più scottanti novità. Scrivere o telefonare a: Stefano Galimberti, via Campaccio 8, 20038 Seregno, tel. 0362/229698.

Software Amiga scambio. Per informazioni scrivere o telefonare a: Paolo Iacobelli, via Aereoporto 15/c, 56100 Pisa (PI), tel. 050/501828.

Cerco urgentemente programmi di comunicazione, in particolare per collegarmi ai videotex standard Prestel. Compro o eventualmente cambio altri programmi, il tutto in MS-Dos. Scrivere a: Mario Renzoli, Piazza N. Tommaseo 9, 50135 Firenze (FI).

Utenti Amiga cerco per scambio software. Per ogni informazione scrivere o telefonare a: Roul Invernizzi, via G. Marconi 26, 20090 Trezzano s/n (MI), tel. 02/4453740 (ore pasti).

Adventure della Infocom (suspect, balchio, sealstaker) compro e cerco inoltre Jinxter con relativo manuale. Per informazioni contattare: Giuliano Galea, via Gradeletto 2, 30071 Mestre (VE), tel. 041/975853 (ore serali).

Programmi Amiga scambio; inviare lista a: Marzio Ventacoli, via Standhal 30, 20144 Milano (MI).

Demodulatore CW-RTTY-AMTOR per C64 - 128 vendo. Cambio programmi per Amiga 500. Guglielmo Bacchetta, Casella Postale 374, 60035 Jesi (AN), tel. 0731/3229-56705-60035.

Cambio programmi e informazioni Amiga 500, scrivere o telefonare a: Emiliano Bagnoli, via Mascagni 39, 41100 Modena (MO), tel. 059/237675.